**The Dissertation Committee for Asad Amin Bawa Certifies that this is the approved version of the following dissertation:**

# Techniques to Increase Compaction of Output Responses with Unknown (X) Values

**Committee:**

Nur A. Touba, Supervisor

Zhigang (David) Pan

Earl Swartzlander

Lizy Kurian John

Muhammad Aater Suleman

# Techniques to Increase Compaction of Output Responses with Unknown (X) Values

**by**

**Asad Amin Bawa**

**Dissertation**

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

## Doctor of Philosophy

**The University of Texas at Austin**

**December 2017**

## Dedication

Dedicated to:

My parents Muhammad Amin Bawa and Farhat Amin Bawa

My wife Nida and our dear children Qasim and Khadija

My sister Bisma and my brothers Muqeet and Salman

My entire extended family and all those who persevere in the pursuit of knowledge

# Acknowledgements

In the name of God, The Most Gracious, Most Merciful. Prayer and blessings be upon the best of creation, my Master Muhammad, and upon his family and companions. First and foremost, all praise and thanks belong to Allah, the Most Merciful, for enabling me to complete this work.

I would like to take this opportunity to thank my loving parents, who inspired and motivated me to pursue this doctoral degree. I sincerely appreciate their tireless efforts and continuous prayers at every stage of my life.

I am indebted to my advisor Dr. Nur Touba for his invaluable mentorship throughout my PhD work. Nur's expertise in the subject, in addition to his very pleasant personality, makes him a great educator. I express my sincere gratitude and consider myself very fortunate to have worked with him.

Additionally, I would like to thank my committee members Dr. David Pan, Dr. Lizy John, Dr. Earl Swartzlander and Dr. Aater Suleman for keeping me on track. Furthermore, I would like to give due credit to the University of Texas at Austin for providing me with the opportunity to learn and pursue my passion for higher education.

Recognition must also be given to my friends and extended family who have been with me through this very rewarding journey, in particular Irfan Bidiwala, Saad Godil, Bilal Janjua, Irfan Waheed, and Kamran Saleem. My friend Tauseef Rab deserves a special mention for meeting up with me on multiple occasions and spending countless hours brainstorming.

I am highly grateful to my wife Nida, for her sincere support. This dissertation could not have been completed without all her sacrifices and constant encouragement.

Last but not least, I would like to acknowledge the most important members of my family, my children Qasim and Khadija, who were my source of inspiration to finish what I started.

# Techniques to Increase Compaction of Output Responses with Unknown (X) Values

Asad Amin Bawa, Ph.D.

The University of Texas at Austin, 2017

Supervisor: Nur A. Touba

Testing requires checking whether the output response of a circuit or system is correct or has an error. Increasingly complex system-on-chip and 3-D integrated circuits require enormous amounts of manufacturing test data. Test compression techniques are widely used to compress the amount of output response data in a way that if an error is present in the uncompacted output response, it will also be present in the compacted output response with only a negligibly small chance of aliasing. Compacting the output response reduces the number of channels needed on the automatic test equipment (ATE) and reduces tester memory requirements. A major challenge for output compaction techniques is dealing with unknown (X) values in the output response which may arise from many sources such as uninitialized memories, analog blocks, tri-states, false paths, etc. While some compactor designs can guarantee observation of errors in the presence of a small number of X's, they may not be sufficient for designs with high X-densities which are becoming increasingly common. This dissertation presents novel advanced techniques to further optimize the handling of X's and scale existing schemes to handle higher X-densities. New designs and techniques will be presented to reduce the control data required to more efficiently handle X's and achieve higher compression with experimental results in the respective sections.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1: Introduction

Because of low manufacturing yields, IC's must be thoroughly tested after they are manufactured to weed out defective parts. Conventional testing involves running automatic test pattern generation (ATPG) software to generate test vectors that target faults and then applying the test vectors to the device-under-test from automated test equipment (ATE). Modern IC's incorporate dedicated design-for-test (DFT) circuitry to allow more efficient testing. One of the basic DFT techniques used is to stitch the sequential elements of the design together into scan chains to allow test patterns from the tester to be shifted in to the circuit-under-test (CUT), and the output response after the test is applied to be shifted back to the ATE. The rate at which test data can be transferred between the tester and the CUT is limited by the bandwidth between the tester and the CUT.

As technology has scaled, the amount of logic implemented in chips has grown faster than the number of chip pins such that the amount of logic to be tested per pin has increased. This has put an increasing stress on the bandwidth between the ATE and CUT. On the other hand, smaller technologies allow for more transistors on SOC which has continued to double every eighteen months. More transistors and the growing density of each technology generation lends itself to higher faults and means more test data on the ATE needs to be transferred to the CUT to test the additional transistors and achieve higher coverage. Consequently, test data volume has increased faster than the bandwidth to the tester resulting in significantly increasing test time. Testing cannot go any faster than the amount of time it takes to get the data from the ATE and test time can be generalized with the equation below [Touba 06]:

$$\textit{Test time} \geq \textit{(amount of test data on tester)} / \textit{(number of tester channels} * \textit{tester clock rate)}$$

## 1.1 Test Compression

Test compression techniques have been widely adopted in the industry to both reduce test time and increase the life of existing ATE by reducing ATE memory requirements. As shown in Figure 1 the main idea of test compression is to add decompressor DFT logic on the CUT to decompress test data from the ATE and Compressor DFT logic to compress the output response before it is transferred back to the ATE. This additional hardware overhead allows data on the tester to be stored in compressed form and provides two main benefits. Firstly, it reduces the memory footprint of the test set on the ATE since it is compressed and extends the life of old testers with lower memory. Secondly since the data on the ATE is transferred in compressed form and decompressed on the CUT, the test time is reduced for a given bandwidth between the ATE and CUT and this is the main benefit of test compression that justifies the hardware overhead.

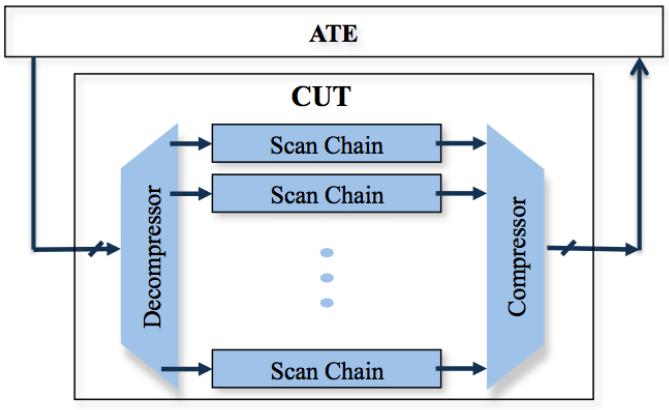


**Figure 1. DFT Architecture**

As can be seen from Figure 1 there are two sides to test compression, the input compression which takes compressed data from the ATE and decompresses on the CUT and the output compression that compresses data on the CUT and transfers back to the ATE. In order to maintain fault coverage, the ATE has to apply a precisely deterministic test set to the CUT and this requires the input compression to be lossless as it must be able to produce all the care bits after decompression.

On the other hand, output compression does not need to be lossless and can lose some data with minimal impact to fault coverage. This enables sequential linear compactors like multi-input-shift-registers or MISRs to be used for output compaction and these can achieve very high compaction. Figure 2 shows an example of a MISR that is compacting data from the six scan chains and storing the results in compressed form. MISRs keeps shifting data from scan chains over time by compressing output responses over multiple clock cycles and have minimal coverage loss due to very low probability of aliasing making them ideal for output compression.

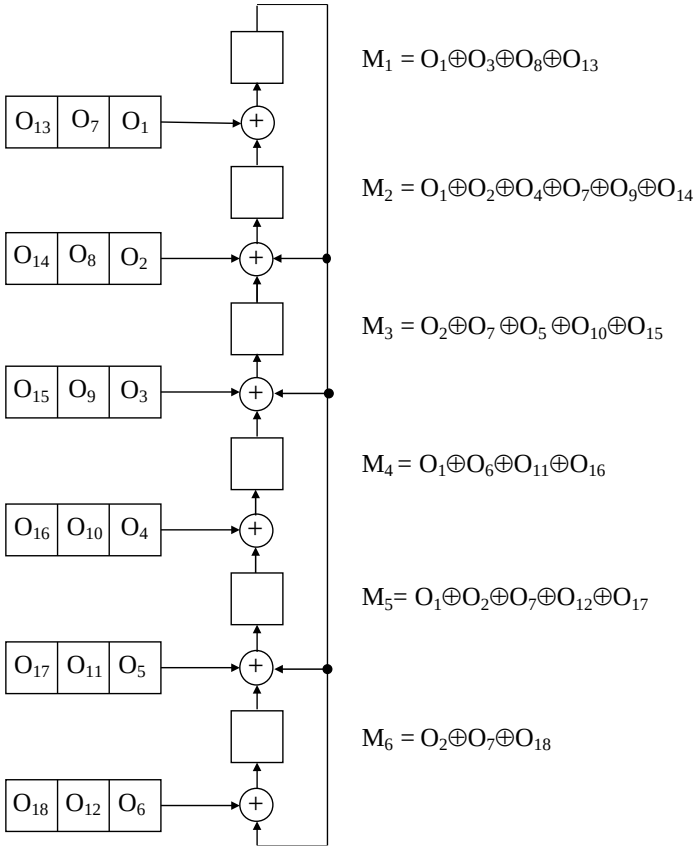


**Figure 2. MISR**

## 1.2 Sources of X's

One major challenge in test compression is the fact that the output response may contain some unknown (X) values, which when compressed can corrupt large portions of the output response such that the ATE can miss errors resulting in lower fault coverage and unacceptable test quality. Figure 3 shows an example of a circuit that can create an unknown (X) during testing. In functional operation, this circuit will be guaranteed to only enable one of the two tri-state drivers, however during testing, pseudo-random patterns are shifted to the CUT from the ATE and if those patterns enable both tri-state drivers, the output driven by them will be an unknown (X).

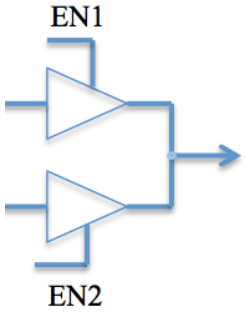


**Figure 3. Example of X generating circuit**

There are many other common sources of unknown values in the output response including uninitialized memories, analog blocks, false paths, and multi-cycle paths. During testing and initial CUT bring up for testing the data in memories on the CUT is not valid and unknown. After scan data is shifted into sequential elements of CUT and a functional capture cycle is applied, data from these memories is also captured into sequential elements being tested. Since the data in the memory is an X, the data captured and shifted out to the ATE is also an X. Similarly, analog blocks are not part of digital testing and any sequential element in the fanout cone of an analog block during testing may be capturing Xs.

The X's generated due to false paths and multi-cycle paths are generated within the CUT that is being tested, similar to the two tri-state drivers shown in Figure 3. In functional operation, a timing path can be guaranteed to be false or known to be multicycle. For example, during an arithmetic multiply operation the functional state machine is designed to wait x-clock cycles before a valid output is expected. During scan testing this may be timed at single cycle, even though the multiplier is given known data, the output of the multiplier after a single cycle will generate a bunch of X's which will get captured and shifted out to the ATE.

As technology shrinks SOCs are getting bigger and more complex and there is increasing interaction between analog blocks and digital ICs. Increasingly complex designs

are lending themselves to a larger number of timing exceptions like false paths and multi-cycle paths and all of these is causing the X's seen during testing to continue to go up.

## 1.3 Handling X's in Output Compaction

While automated test equipment (ATE) can be programmed to ignore certain bits in the output response, in test compression and BIST, output compaction is employed where multiple output response bits are exclusive-ORed (XORed) together on-chip. As shown in Figure 4, if an X bit is XORed with a non-X bit, then observation of the non-X bit is lost. Loss of observation reduces the ability to perform diagnosis, reduces coverage of non-modeled faults, and can result in more test patterns needing to be applied to achieve the same fault coverage (i.e., test pattern inflation). The amount of observation that is lost scales directly with the amount of compression employed thereby making this problem a major impediment to achieving high amounts of test compression. Compounding the problem is the fact that the ratio of X-values to non-X values (i.e., the X-density) is trending higher in each successive generation of technology.

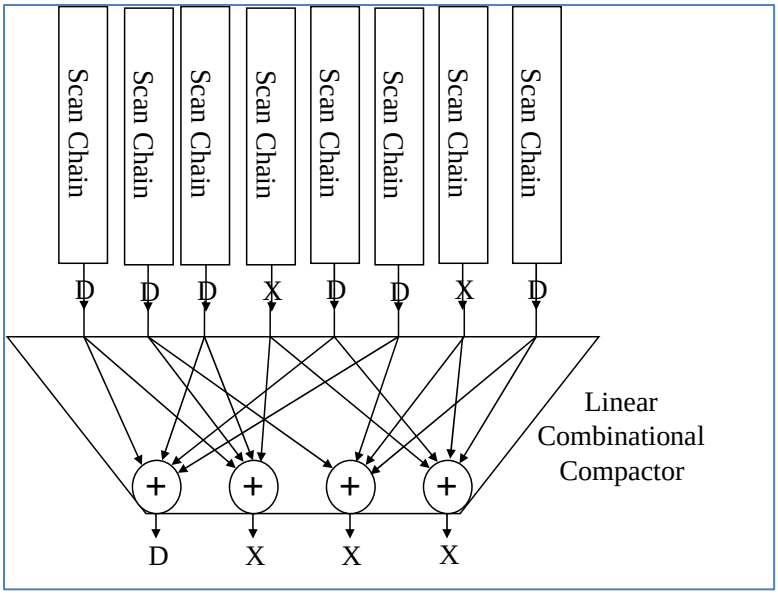


**Figure 4. Unknowns Corrupting output of Linear Combinational Compactor**

A number of techniques have been developed to handle X's in the output response. One approach is to modify the circuit-under-test (CUT) to eliminate the sources of X-values. This involves blocking sources of X within the circuit by inserting design-for-testability (DFT) hardware to prevent Xs from propagating into scan cells [Wang 06]. Another approach, which does not require modifying the CUT, is X-masking which masks out X's at the input to the compactor. Mask control data is used to specify which scan chain outputs should be masked during specific clock cycles. Many schemes for X-masking hardware design and mask control data compression have been developed [Barnhart 01], [Pomeranz 02], [Chickermane 04], [Volkerink 05], [Chao 05], [Tang 06], [Rajski 08] and [Mrugalski 09]. A third approach is to use an X-tolerant compactor which can compact an output stream that contains X's without the need for X-masking. X-tolerant compactors have been developed based on linear combinational compactors [Mitra 04], [Sharma 05], [Wohl 07a, 07b], finite memory compactors [Wang 03], [Rajski 05, 06], [Gizdarski 10], and X-canceling MISRs [Touba 07], [Yang 12] and [Chung 12].

## 1.4 Motivation

All of the previous techniques struggle to handle high X-densities while still maintaining high test compression. As chips continue to increase in size and complexity, the number of X's is expected to continue increasing with the trend towards greater use of memories, analog blocks, and complex multi-cycle paths. Compacting output streams that have X's is a major issue for test compression and also built-in self-test (BIST). Solutions to efficiently handle X's with minimal modification of the circuit-under-test (CUT) are needed. This is the motivation of this work, and this dissertation presents novel designs and techniques to increase the efficiency of handling X's and improve output compaction for higher X densities. The contributions of this dissertation are:

- A novel technique for increasing compaction in the presence of high X-densities using partial masking in X-chains together with an X-canceling MISR. An X-Canceling MISR [Touba 07] provides the ability to tolerate unknowns (X's) in the output response with very little loss of observability of non-X values. When the density of X's is low, an X-Canceling MISR is extremely efficient as the number of control bits depends only on the total number of X's in the output response. However, for higher X-densities, an X-Canceling MISR becomes less efficient. The use of X-chains enables an X-cancelling MISR to be used for higher X-densities but requires a careful selection of the number of X-chains. This work is described in Chapter 2 and was published in [Bawa 12].
- A method for achieving greater compression of X-masking control data for a given number of tester channel by performing dynamical channel allocation. One approach for handling X's which does not require modifying the circuit-under-test (CUT) is to use X-*masking* in which X's are masked out at the input to the compactor, e.g., a multiple-input signature register (MISR). By using dynamic channel allocation together with test vector ordering more free variables are created and this spare capacity is used to improve observability. This work is described in Chapter 3 and was published in [Bawa 13].
- A new approach of using a combinational rotator between the scan chains and a combinational compactor to scale for higher X-densities without the need for masking. For low X-densities, techniques that use sequential linear compactors such as MISRs or convolutional compactors, can typically achieve higher amounts of compression than combinational compactors. For high X-densities, combinational compactors become more efficient in preserving observability thereby reducing test pattern inflation so as to achieve a higher overall amount of test compression. This work is described in Chapter 4 and was published in [Bawa 15].
- A novel and completely new methodology for designing a combinational output compactor based on using an input rotator that is able to maintain high observability even in the presence of high X-densities while still achieving high compaction ratios.

The key idea here is to maximize the separation of bits that are compacted after rotation. This work is described in Chapter 4 and was published in [Bawa 17].

## 1.5 Dissertation Organization

This dissertation is organized as follows. In Chapter 2 masking techniques are presented to improve the efficiency of a sequential linear compressor (MISR) which is followed by Chapter 3 that presents new ways of compressing masking control data to achieve higher compression. Chapter 4 presents a new idea of using a rotator to improve observability of X-tolerant combinational compactors and in Chapter 5 a novel linear combinational compactor is presented that uses the rotator presented in Chapter 4 to achieve significantly higher observability and handle a much larger X-density. Experimental results and conclusions are presented in the respective chapters.

# Chapter 2: Partial Masking in an X-Canceling MISR

## 2.1 The Problem

The advantage of an X-canceling MISR (the operation of which will be described in Sec. 2.3 ) is that it is very precise in canceling out the X's, losing observation of very few of the non-X values. Thus, it retains both the modeled and non-modeled fault coverage of the original test set. Conventional X-masking approaches which mask out entire scan chains or entire scan slices are more blunt approaches which lose observation of a considerable number of scan cells in the process of masking out X's. This results in more test vectors needing to be applied to achieve the same fault coverage and impacts non-modeled fault coverage. When the X-density is low, an X-canceling MISR is very efficient and requires fewer control bits than conventional X-masking approaches. However, as the X-density increases, the control bits for X-canceling begins to exceed those required for X-masking, and it becomes less efficient.

## 2.2 Related Work

Some previous work has looked at improving the efficiency of an X-canceling MISR by trying to reduce the number of X's that reach it. In [Datta 11], circular registers are used to stack X's on top of each other so they can be treated as a single X when performing X-canceling. In [Ramdas 12], a toggle-masking scheme is used to mask consecutive bits of X's at the output of all scan chains. This requires $\log_2(n+1)$ dedicated control bit channels from the tester for n scan chains and significant design overhead (n XOR gates, n MUX gates, n flip-flops and a $\log_2(n+1)$ by $(n+1)$ decoder). The proposed scheme performs a much lower cost partial X-masking requiring only one dedicated control bit channel from the tester and only one AND gate per chain for a subset of scan chains.

It has been observed by researchers [Czysz 10] and [Wohl 10], that there is a lot of locality in the scan cells where X's are captured. A relatively small percentage of the scan cells capture the vast majority of the X's that are generated. In [Wohl 08], the idea of

stitching together the scan cells that capture the most X's into a small number of "X-chains" was proposed in the context of combinational compaction. The modes of the combinational compactor were designed so that the X's in the X-chains could be tolerated with less cost. This chapter proposes an approach that uses the idea of X-chains in the context of an X-canceling MISR. In the proposed approach, a partial X-masking scheme is used for the X-chains to mask the vast majority of the X's at very low cost. The partial X-masking scheme involves analyzing the output response in the X-chains and avoiding masking slices when there will be loss of fault detection. In the slices where the X-chains are not masked, any X's that appear are allowed to pass through to the MISR, which will be referred to here as "X-leaking." In a conventional X-masking scheme, X-leaking is not tolerable because it will corrupt the MISR signature. Consequently, over masking ends up being done to ensure that there is no X-leaking. This results in loss of observability and consequently loss of fault detection. More ATPG vectors end up being used to achieve the desired modeled fault coverage. With an X-Canceling MISR, X-leaking is not a problem because the X's can be canceled out in the MISR signature. The responses in the scan slices that are not part of the X-chains also go into the X-canceling MISR where any X's that appear are canceled.

The proposed approach exploits the fact that the number of control bits required to cancel X's in an X-canceling MISR depends only on the total number of X's. So, it is very efficient to handle any residual X's that don't get canceled in the X-chains. By masking the vast majority of the X's with the low cost partial masking in the X-chains, the control bits required to handle the remaining X's that get into the X-canceling MISR are greatly reduced.

## 2.3 Review of X-Canceling MISR

This section gives a brief overview of the operation of an X-canceling MISR. A more detailed explanation can be found in [Touba 07].

Assume the output response has been captured in the scan chains after applying a test vector. The value in each scan cell is represented with a symbol. An example is shown in Figure 5. Once the output response has been shifted in to the MISR, the final MISR signature can be expressed in terms of the symbols through symbolic simulation. Each MISR bit is represented by a linear equation of the scan cell symbols. Figure 5 illustrates this symbolic representation. The final value of the top bit of the MISR is $X_1 \oplus O_3 \oplus O_8 \oplus O_{13}$, where $X_i$ denotes an X value and $O_i$ indicates a non-X value.

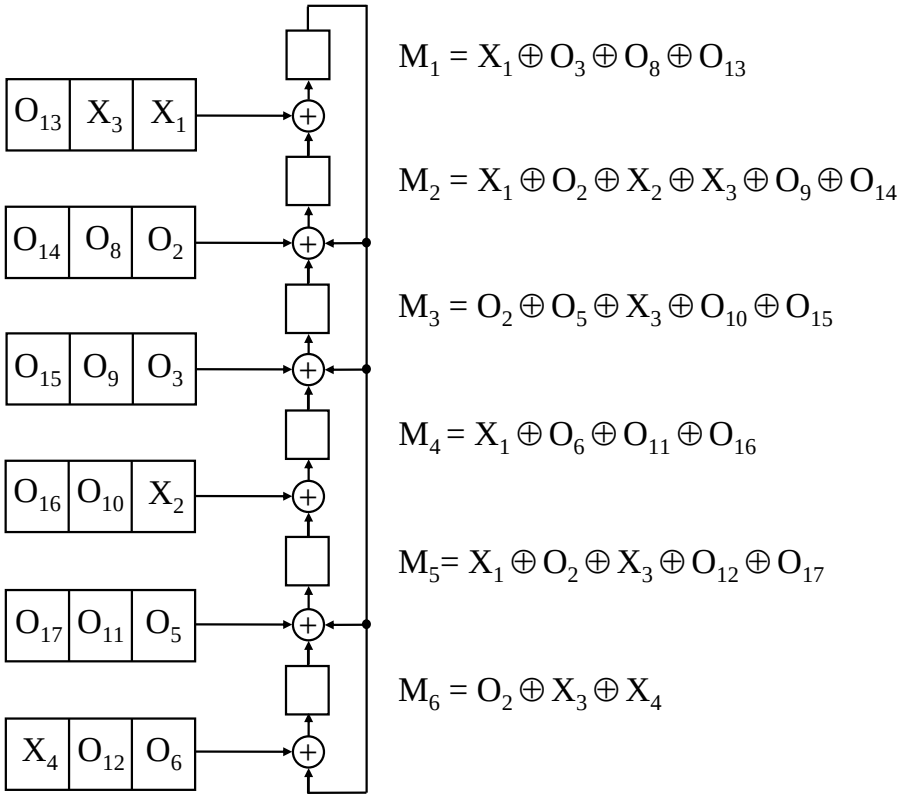


**Figure 5. Example of Symbolic Simulation of MISR**

$$
\begin{aligned}
M_1 &= X_1 \\
M_2 &= X_1 \oplus X_2 \oplus X_3 \\
M_3 &= X_3 \\
M_4 &= X_1 \\
M_5 &= X_1 \oplus X_3 \\
M_6 &= X_3 \oplus X_4
\end{aligned}
\quad \rightarrow \quad
\begin{bmatrix}
1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix}
$$

**Figure 6. Linear Equations for MISR**

$$
\begin{bmatrix}
1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix}
\begin{matrix}
M_1 \\ M_2 \\ M_3 \\ M_4 \\ M_5 \\ M_6
\end{matrix}
\quad \xrightarrow{\text{Gaussian Elimination}} \quad
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\begin{matrix}
M_1 \\ M_1 \oplus M_2 \oplus M_3 \\ M_3 \\ M_3 \oplus M_6 \\ M_1 \oplus M_3 \oplus M_5 \\ M_1 \oplus M_4
\end{matrix}
$$

**Figure 7. Gauss-Jordan Elimination of MISR Equations**

The focus here is on the unknown values, so each MISR bit equation can be reduced to a linear combination of the X values by assigning *0* to each non-X value without loss of generality. These linear combinations can be expressed in the form of a matrix as shown in Figure 6. Each entry in the matrix has a *1* if the MISR bit corresponding to the row depends on the X corresponding to the column.

If the number of columns is less than the number of rows, i.e., the number of X's is less than the MISR size, then some row combinations will be linearly dependent. Gauss-

Jordan elimination [Cullen 97] can be performed on the matrix in Figure 6 to identify the linearly dependent combinations of rows as illustrated in Figure 7. The last two rows in Figure 7 have all *0*s and this indicates combinations of MISR bits in which all the X's cancel out. The first all-*0* row corresponds to $M_1 \oplus M_3 \oplus M_5$. This implies that XORing MISR bits $M_1$, $M_3$, and $M_5$ generates an "X-canceled" signature bit which depends only on scan cells that captured non-X values as shown below:

$$M_1 \oplus M_3 \oplus M_5 = O_3 \oplus O_5 \oplus O_8 \oplus O_{10} \oplus O_{12} \oplus O_{13} \oplus O_{15} \oplus O_{17}$$

The values of these X-canceled MISR bit combinations are deterministic and can be predicted through simulation. Therefore, during test, they can be compared with their fault-free values in order to detect errors.

The architecture of an X-canceling MISR is shown in Figure 8. The MISR captures response across many clock cycles and may span multiple test vectors until the MISR fills up with X's. The MISR signature is then processed by selectively XORing linearly dependent combinations of MISR bits in terms of the X's to generate an X-free output response to send to the tester. The error coverage can be made arbitrarily high by generating and checking a sufficient number of X-canceled output responses. The probability of not detecting an error drops by a factor of 2 for each X-canceled combination that is checked. If $q$ X-canceled combinations are checked, then the error coverage for it will be $1-2^{-q}$. So, if $q$=7, then the error coverage will be 99.2%, and each MISR signature can capture up to $(m-7)$ X's where $m$ is the size of the MISR.

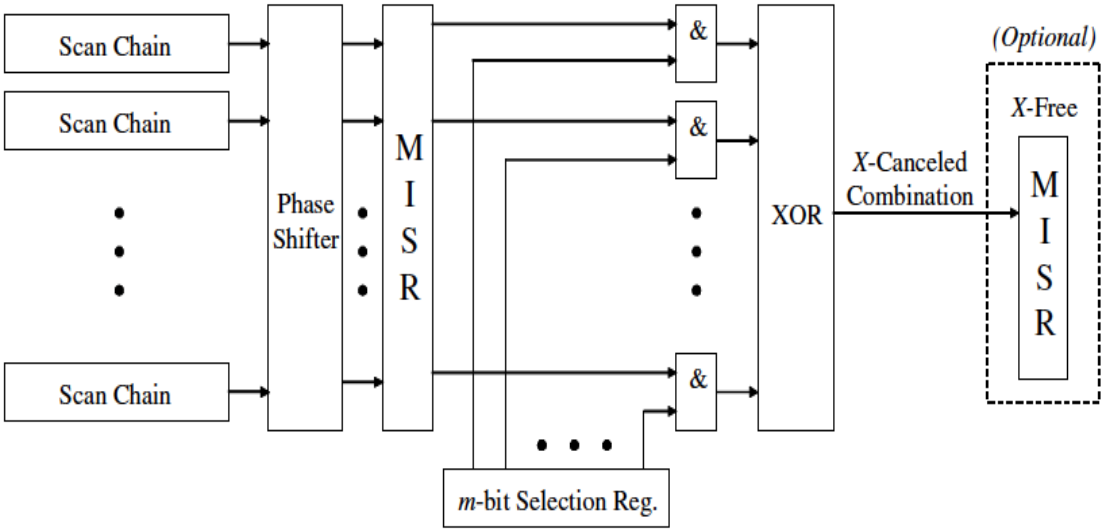


**Figure 8. X-canceling MISR Architecture**

## 2.4 Partial Masking in X-Chains to Increase Compaction for an X-Canceling MISR

The proposed method describes a very effective approach for using an X-Canceling MISR for designs with high X-density. It utilizes the idea of stitching together scan cells that capture the largest number of X's into "X-chains" as was proposed in [Wohl 08]. In the proposed approach, a partial X-masking approach is used for the X-chains to eliminate the vast majority of the X's at very little cost in terms of control bits. Only the X's coming from the scan cells not in the X-chains plus X's that are left unmasked in the X-chains need to be handled by the X-canceling MISR thereby significantly reducing the total number of control bits required. Experimental results show an order of magnitude improvement in the output compaction can be achieved.

A block diagram of the proposed scheme is shown in Figure 9. As can be seen, there is a mask bit that is set to 0 to mask the data coming out of X-chains to a known value which is 0. The masking is done on a cycle-by-cycle basis. This mask bit is set to 0 in all cycles except when one of the data bits coming from the X-chains has a value that must be observed to ensure detection of faults. This is determined by keeping track of which scan

cells the fault effects propagate to during fault simulation and marking them as *D*'s. When performing X-masking, the *D*'s should not get masked. One tester channel is used to drive the X-chain mask control bit and masks all the X-chains in clock cycles when no *D*'s occur, but does not mask X-chains in cycles when there are one or more *D*'s. When the X-chains are not masked, there is a possibility of an X passing through to the MISR which results in "X-leaking". In conventional X-masking applications, X-leaking cannot be tolerated, but in this scheme with an X-canceling MISR, they can be canceled out in the MISR signature.

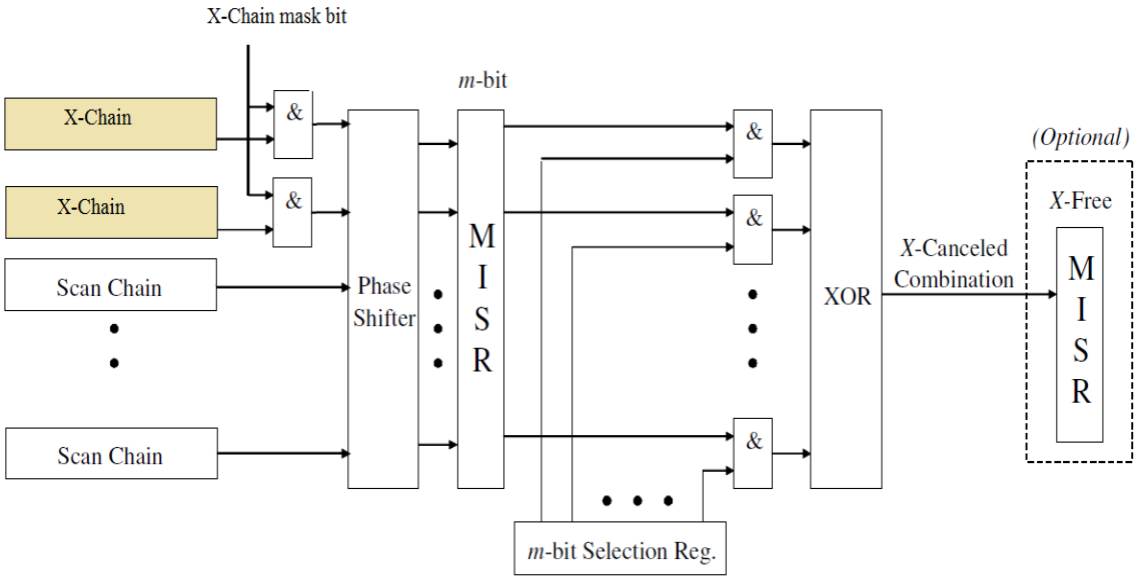


**Figure 9. X-canceling MISR Architecture with X-Chains**

The advantage of using the proposed scheme compared to just using a conventional X-canceling MISR is that it takes fewer control bits to mask out X's in the X-chains than it does to cancel them out in the MISR signature. One control bit is used in each clock cycle, and since the X-chains have a high density of X's, many X's get masked out. This

significantly reduces the number of X's that have to be canceled out in the X-canceling MISR.

The overall effectiveness of this scheme will depend on how many X-chains are used, since the greater the number of X-chains the greater the chances that one of them will have a *D* in a given scan slice and thus result in more X-leaking. On the other hand, if the number of X-chains are too small, then there will be more X's in the regular scan chains and that will also lead to more X's in the MISR. The ideal number of X-chains that are used will thus be design dependent. Selecting the optimal number of X-chains is discussed in the next section.

## 2.5 Selecting Number of X-Chains

The optimum number of X-chains depends on the frequency of X's and the frequency of *D*'s. Table 1 shows experimental results for a circuit where different numbers of X-chains were tried. The first column shows the number of X-chains. The second column shows the percentage of the X's that ended up being captured in the X-chains. As the number of X-chains increases, the percentage of X's captured in the X's chains increases until it reaches 100% with 36 X-chains. The third column shows the total percentage of X's masked in the X-chains (which obviously cannot be greater than the percentage of X's captured in the X's chains). The reason why the percentage of X's masked is less than the percentage captured in the X-chains is because of the presence of *D*'s in the X-chains which prevent masking in some clock cycles. As the number of X-chains increases, the chance of a *D* appearing during a given clock cycle goes up and consequently the amount of X-leaking also goes up. As can be seen, the percentage of X's that are masked peaks at 12 X-chains. When the number of X-chains is increased to 18, even though the total number of X's in the X-chains increases from 95.6% to 99.0%, the X's masked goes down because the *D*'s cause more X-leaking. The last column shows the total number of control bits which is equal to one bit per clock cycle for the X-chain mask control bit plus the control bits required for canceling out the X's that get into the X-

canceling MISR. The number of control bits is minimized for 12 X-chains because at that point the percentage of X's masked is maximized meaning less X-canceling is required. This information is also shown graphically in Figure 10 where the total control bits is plotted on the y-axis with the number of X-chains in the X-axis. As can be seen, as the number of X-chains is increased, initially the total control bits is reduced until it is minimized at some point beyond which it begins to increase as additional X-chains are added.

This same type of curve was observed for all circuits in which experiments were performed. Selecting the optimal number of X-chains involves performing simulations to find the number of X-chains where the curve minimizes.

It is possible to partition the X-chains and use multiple tester channels to bring in more control signals to perform partial X-masking on each of the partitions of X-chains. While this will reduce X-leaking, the gain in terms of overall control bits is not substantial enough to offset the cost of using the additional tester channels. As can be seen in Table 1, 89.2% of X's is already masked with 12 X-chains. So, the maximum possible improvement for more masking is limited to only 10.8% which is simply not enough to make the use of an additional tester channel to support more X-chains to be worthwhile.

**Table 1 Experimental Results for Ckt-A Using Different Numbers of X-chains**

| Num. X-Chains | X's in X-Chains | X's Masked | Total Control Bits |
|---|---|---|---|
| 4 | 58.7% | 58.1% | 22.3M |
| 8 | 84.5% | 81.4% | 10.7M |
| 12 | 95.6% | 89.2% | 6.8M |
| 18 | 99.0% | 87.3% | 7.8M |
| 36 | 100% | 73.6% | 14.6M |

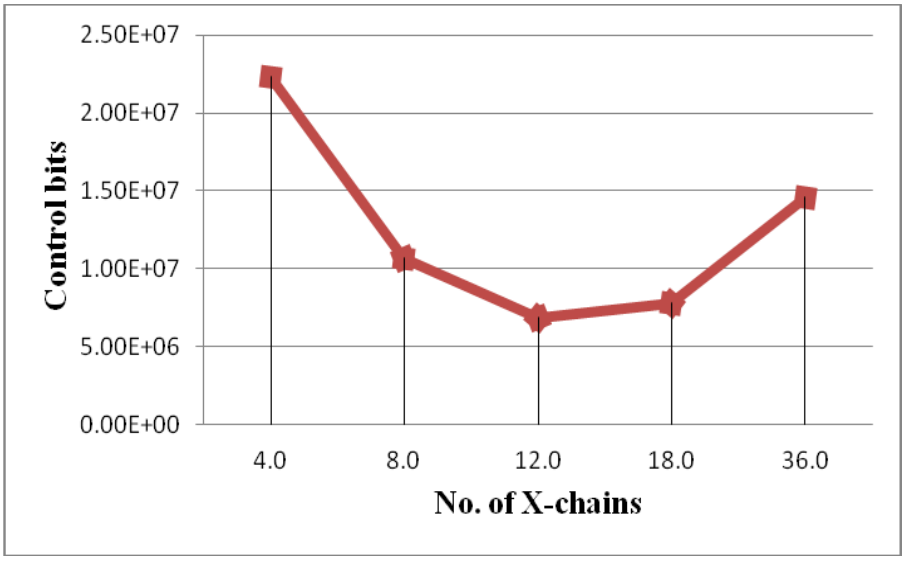


**Figure 10. Reduction in Control Bits versus X-Chains for Ckt-A with 1% D's**

## 2.6 Experimental Results

Experiments were performed on three industrial circuits with different X-densities to evaluate the proposed method of using partial masking in X-chains in conjunction with an X-canceling MISR [Touba 07]. A 256 bit MISR was used, however, as shown in [Touba 07], the total number of control bits is relatively independent of the size of the MISR, so different size MISRs would show similar improvements. The circuits themselves were not available, so fault simulation was not used to mark the exact locations of the *D*'s. Instead, *D*'s were randomly injected assuming different percentages of *D's*.

Table 2 shows results for three different circuits for different numbers of *D*'s and X-chains. The X-density for each circuit (which is the percentage of X's) is shown in the first column. The second column shows the total number of bits stored on the tester for using an X-canceling MISR as described in [Touba 07]. The third column shows the number of X-chains that was used. Three different numbers of X-chains were tried for each circuit. The next columns show the results for the proposed scheme assuming 0.5%

*D*'s, 1% *D*'s, and 2% *D*'s. For each case, the total number of bits stored on the tester is shown and the improvement factor versus using an X-canceling MISR by itself is shown.

**Table 2. Experimental Results for Proposed Partial Masking in X-Chains for Different Designs**

| Circuit | X-Canceling [Touba 07] Bits | Num. X-Chains | 0.5% *D*'s | | 1% *D*'s | | 2% *D*'s | |
|---|---|---|---|---|---|---|---|---|
| | | | Bits | Improve Factor | Bits | Improve Factor | Bits | Improve Factor |
| Ckt-A X-density = 2.4% | 49.97M | 10 | 5.1M | 9.7 | 8.2M | 6.1 | 10.3M | 4.8 |
| | | 12 | 5.2M | 9.6 | 6.8M | **7.3** | 9.8M | **5.1** |
| | | 15 | 4.6M | **10.8** | 6.8M | 7.3 | 11.0M | 4.6 |
| Ckt-B X-density = 2.7% | 21.33M | 2 | 8.5M | 2.5 | 8.5M | 2.5 | 8.5M | 2.5 |
| | | 4 | 3.6M | 5.9 | 3.6M | 5.9 | 4.0M | 5.3 |
| | | 6 | 1.9M | **11.2** | 2.3M | **9.3** | 3.0M | **7.1** |
| Ckt-C X-density = 0.5% | 5.2M | 1 | 1.9M | 2.7 | 1.9M | 2.7 | 1.9M | 2.7 |
| | | 2 | 1.5M | **3.5** | 1.5M | **3.5** | 1.5M | **3.5** |
| | | 4 | 1.5M | 3.5 | 1.6M | 3.3 | 1.7M | 3.1 |

As can be seen in Table 2, a significant improvement in compression is achieved with the proposed scheme for both low and high X-density circuits. The greatest improvement is for Ckt-*B* which has the highest X-density. For Ckt-*C*, even though the X-density is very low it still gives significant improvement in compression. It can be expected that as the X-density increases, the proposed scheme will provide increasingly larger improvement factors.

## 2.7 Summary

An X-canceling MISR was shown in [Touba 07] to be very efficient for small X-densities. The scheme proposed in this chapter exploits this fact by combining X-canceling with partial X-masking to handle high X-densities. The result of masking X's in X-chains is that the MISR has a lot fewer X's and requires less cancelling which in turn means fewer control bits are needed. Because the proposed method can handle X-leaking, it is able to achieve high compression while still providing very precise masking where very little observation of non-X values is lost. This results in fewer test vectors and hence better test vector compression, output response compression, and test time. Since the test data bandwidth is always fully utilized with an X-canceling MISR, the test time reduction will scale directly with the reduction in control bits.

# Chapter 3: Compression of X-Masking Control Data via Dynamic Channel Allocation

## 3.1 The Problem

One approach for handling X's which does not require modifying the circuit-under-test (CUT) is to use X-*masking* in which X's are masked out at the input to the compactor, e.g., a multiple-input signature register (MISR). In Chapter 2 partial X-masking was used on X-chains in combination with an X-tolerant compactor but for compactors that cannot tolerate unknowns, X-masking must be applied to all scan chains that can have an X. The key issue for X-masking is how to select which bits to mask. It would be ideal if only the X values were masked and all non-X values were not masked, but the amount of mask control information required for this level of precision is prohibitive. Consequently, some loss of observability is necessary in order to sufficiently compress the mask control information. This chapter will present novel ideas to improve the compression of X-masking control data.

## 3.2 Related Work

A number of schemes for performing X-masking have been proposed. One approach is to mask the output of scan chains that contain X's during a whole scan out [Barnhart 01] and [Rajski 02]. If $m$ is the number of scan chains, this approach would require $m$ control bits (one per scan chain to indicate if it should be masked or not masked) for each test vector. In this approach, the amount of mask control data is small, but observability is lost for all scan cells in each masked scan chain which can be problematic if some fault is only observable in a particular scan chain.

To avoid losing observability of entire scan chains, another approach is to select which scan chains to observe in each shift cycle. To reduce the number of control bits required in each shift cycle in this case, a small number of modes can be defined in which certain combinations of scan chains can be either masked or observed [Chickermane 04]

and [Wohl 07b]. In this case, the number of control bits required for each shift cycle is $\log_2$ of the number of modes.

Some techniques have been developed to exploit correlations in the location of X's across output responses to reduce the amount of mask control data. In [Wohl 08], the scan cells that capture the most X's are stitched in "X-chains" which allows the masking modes to be better optimized [Wohl 08]. In [Wohl 10] and [Czysz 10], the mask control data for output responses with similar X locations are merged together to reduce the amount of mask control data loaded from the tester.

Mask control data can be compressed by using a sequential linear decompressor such as a linear feedback shift register (LFSR) or ring generator [Mrugalski 04]. In [Naruse 03], the mask control data on a per-shift basis is compressed using LFSR reseeding [Könemann 91]. A drawback of this approach is that since the output of an LFSR is 1 and 0 roughly 50% of the time, half of the non-X scan cells get masked as well, so overall observability is reduced by 50%. In [Volkerink 05], multiple stages of an LFSR are ANDed together to reduce the probability of masking non-X values and hence increases observability. In [Wang 08a], a minimal set of scan cells that need to be observed to detect all faults is determined by looking at where the fault effects for each fault propagate during the automatic test pattern generation (ATPG) process. This information is used to simplify the linear equations when performing LFSR reseeding to obtain better compression of the mask control data. Further optimizations for LFSR reseeding are described in [Wang 08b] by using group masking. In [Czysz 10] and [Wohl 10], sequential linear decompressors are used to load a shadow register which holds the mask control data. In this case, when the same masking pattern can be used for consecutive shift cycles, it is not necessary to reload the shadow register. It only needs to be loaded when the masking pattern changes.

Note that when using sequential linear decompressors, the amount of compressed data that is required is proportional to the number of specified bits that need to be generated. Each bit stored on the tester is a free variable that can be assigned either a 0 or 1. To generate a certain set of specified bits at the output of a sequential linear decompressor, a linear equation needs to be solved for each specified (i.e., care) bit. In order to solve the

system of linear equations, more free variables than care bits are required. The number of free variables that need to be delivered to the decompressor is proportional to the number of care bits it needs to generate. In the case of using a sequential linear decompressor to generate mask control data, the number of free variables is proportional to the number of X's in the output response. The number of X's can vary considerably from one output response to the next. The same issue is present when using a sequential linear decompressor for compressing test cubes (i.e., test vectors in which the unassigned inputs are left as don't cares). The number of free variables required to encode a test cube is proportional to the number of care bits in the test cube, which again can vary considerably from one test cube to the next. The tester channels deliver free variables in a steady stream each clock cycle. The number of tester channels and clock cycles used to decompress either a test cube or mask control data is set based on the maximum number of care bits that need to be generated. Hence the compression achieved depends on the worst-case test cube or mask control data.

It is not feasible to use the same sequential linear decompressor for encoding both test cubes and mask control data because when encoding a test cube, the values in the non-care bits cannot be determined until the linear equations are solved. However, without knowing the values of non-care bits in the test cube, it is not possible to determine the presence and location of the X's in the output response. For this reason, separate decompressors have to be used for the test cubes and the output response. Once the test cubes are encoded using the *test cube decompressor* and the fully specified decompressed test vectors are known, then the output response can be determined and the required mask control data can be encoded with the *mask decompressor*. Existing state-of-the-art techniques, e.g., [Czysz 10] and [Wohl 10], are based on having separate decompressors as described above.

Note that the scan out of an output response is done concurrently with the scan in of the next test vector. The free variables that are required to encode a test cube need to be delivered at the same time as the free variables that are required to encode the mask control data for the output response of the previously applied test cube.

## 3.3 Overview of the Proposed Scheme

The idea proposed in this chapter is that rather than the conventional approach of having a fixed number of tester channels feeding the test cube decompressor and a fixed number feeding the mask decompressor, the number of channels feeding each is dynamically adjusted. The key to making this very efficient is careful ordering of the test cubes. Test cubes with larger numbers of care bits requiring more tester channels to bring in free variables are placed after test cubes whose output response has smaller numbers of X's requiring mask control data with smaller number of care bits and hence needing fewer tester channels to bring in free variables. On the flip-side, test cubes with smaller numbers of care bits are placed after test cubes whose output response has larger numbers of X's. By balancing the number of free-variables needed by each decompressor, the worst-case total aggregate number of free variables needed for decompressing any test cube and output response pair can be minimized. This allows greater compression for a given number of total tester channels that are available from the tester.

To illustrate the improvement in encoding efficiency that is possible with the proposed approach, consider the following example. Suppose the worst-case number of free variables needed to encode any test cube was 1000, and the worst-case number of free variables needed to encode the mask data for any output response was 600. If there were 16 tester channels, then the conventional approach would be designed with 10 channels feeding the test cube decompressor and 6 channels feeding the output response decompressor. In this scenario, 100 clock cycles would be used to decompress each test cube so that enough free variables are supplied to the decompressors to encode the worst-case test cube and worst-case mask data. With the proposed approach, the worst-case test cube needing 1000 free variables would get matched with the output response requiring the fewest free variables, let's say it was 100 for example. When decompressing that test cube, 14 channels could be allocated for the test cube decompressor and 2 channels could be allocated to the mask decompressor. In that case, only $\lceil 1000/14 \rceil = 72$ clock cycles would

be sufficient to provide enough free variables for both the test cube decompressor and mask decompressor (which would receive 2x72=144 free variables). For the worst-case mask data needing 600 free variables, perhaps it could be matched with a test cube needing let's say 400 free variables, then when decompressing that test cube, 9 channels could be allocated to the mask decompressor and 7 channels could be allocated to the test cube decompressor. In that case, only $\lceil 600/9 \rceil = 67$ clock cycles or more would be sufficient to provide enough free variables for both the mask decompressor and test cube decompressor (which would receive 7*67=469 free variables). Looking across all test cube and mask data pairs, if the worst-case number of clock cycles needed turned out to be 72, then that would determine the number of clock cycles used when decompressing all test cube and mask data pairs. The improvement in compression achieved by the proposed approach in this example would be (100-72)/72 = 39%. Both the test time and the tester memory usage would be reduced by 39%.

The basic idea in this chapter can be applied on top of any of the existing schemes that use sequential linear decompressors for compressing masking data to achieve greater compression. For simplicity and without loss of generality, the approach is described here using a basic structure of having a sequential linear decompressor feeding masking logic on a per-shift basis similar to what is used in [Naruse 03] and [Volkerink 05]. Adding the optimizations described in other papers (e.g., [Wang 08a, 08b]) or adapting it for other architectures (e.g., Czysz 10]) is straightforward.

One technique for boosting observability that is proposed here uses the idea from [Volkerink 05] of ANDing together the outputs of multiple stages of a sequential linear decompressor to reduce the probability of non-X values getting masked. When multiple stages are ANDed together, it increases the number of free variables needed to encode the mask control data for eliminating the X's. However, it may often be the case that more free variables can be supplied to the mask decompressor without impacting the compression. In other words, if it is determined, as in the example mentioned earlier, that say 72 clock cycles need to be used for the worst-case decompression, there are likely many other test cubes that need much less than the worst-case number of free-variables, so there is

essentially an excess number of free-variables supplied versus the number needed for many test cubes. These excess free variables can be used to increase observability of non-X values by ANDing together multiple stages of the decompressor when producing the mask control data. This extra observability essentially comes at no extra cost in terms of test time or additional data on the tester. Additional observability helps to reduce test vector count and detect non-modeled faults.

## 3.4 Dynamic Channel Allocation

The proposed scheme is based on the idea of dynamically adjusting the number of tester channels that are used to feed the test cube decompressor and the masking decompressor on a per test cube basis. The hardware for implementing this is shown in Figure 11. There are $b$ channels coming from the tester, and each sequential linear decompressor has $b$ injector inputs. At the start of decompressing each test cube, in the very first clock cycle, the data coming from the tester channels is loaded into the control register $q$. The binary number stored in register $q$ is the number of tester channels to use for the test cube decompressor. This number is fed to a selector which selects $channel_1$ through $channel_q$ and masks off $channel_{q+1}$ through $channel_b$ for the test cube decompressor by ANDing them with 0. It does the opposite for the masking decompressor, i.e., $channel_1$ through $channel_q$ are masked and $channel_{q+1}$ through $channel_b$ are passed through. After the first clock cycle in which register $q$ is loaded, in all subsequent clock cycles, the tester channels are used to inject free variables into the decompressors based on the channel allocation determined by $q$.

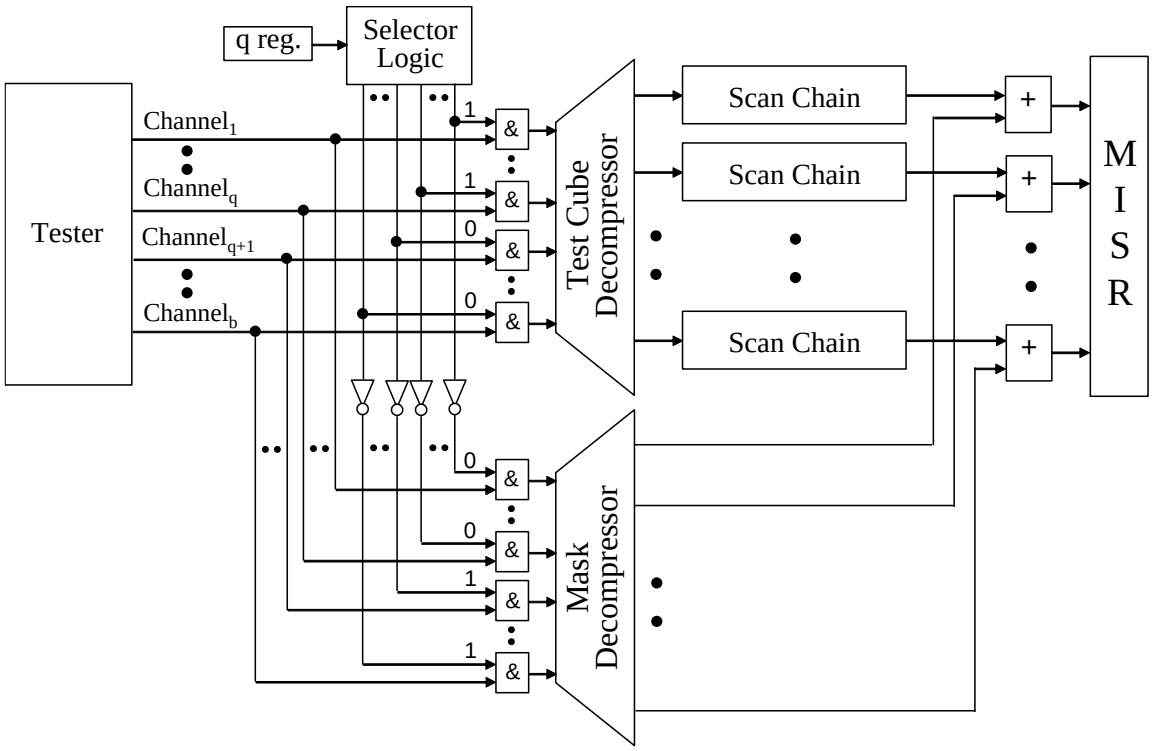


**Figure 11. Proposed Dynamic Channel Allocation Scheme**

Using this hardware, it is now possible to select the channel allocation when decompressing each test cube. The minimum number of channels needed to encode each test cube is determined by first estimating how many free variables will likely be needed to solve the linear equations based on the number of care bits in the test cube. The corresponding number of channels needed to get that number of free variables is then calculated. Then a linear equation solver is used to check if an encoding solution is possible for the estimated number of tester channels [Krishna 01] and [Wang 06]. If so, then the number of channels is incrementally reduced as long as a solution can be found until the smallest number of channels that still produces a solution is identified. If the initial estimate of the number of channels was too low to begin with, then the number of channels is incrementally increased until a solution is found. Once the minimum number of channels needed for encoding the test cube is found, then that is the value of *q* that will be used when

decompressing that test cube, and it will not be changed regardless of which output response is matched with this test cube. This allows the fully specified decompressed test vector to be computed and simulated to obtain the output response and the location of any unknown X's that are produced. These X values must be masked and hence create care bits in the mask control logic. Given the care bits in the mask control logic, the same process of using a linear equation solver as was used for the test cubes can be performed to find the minimum number of channels needed to encode the mask control data for each output response.

Given the number of channels used to encode each test cube, and the minimum number of channels needed to encode the mask control data for each output response, the order for applying the test cubes is selected. As mentioned in section 3.2 , the output response for the previous test cube is scanned out concurrently with the scan in of the current test cube. Hence the order in which the test cubes are applied determines which output responses are matched with which test cubes. The procedure for ordering the test cubes to minimize the total number of tester channels required (and hence maximize compression) is as follows.

For the very first test cube that is applied, the scan out is ignored, so there is no masking data required. The test cube requiring the maximum number of channels for decompression can be applied first. By the same token, no test cube is scanned in when the very last output response is scanned out, the test cube whose output response requires the most channels can be applied last. The rest of the test cubes are ordered such that test cubes requiring the most channels are matched with the output response requiring the fewest channels. A small example to illustrate the test cube ordering is shown in Figure 12. There are 5 test cubes which are reordered to match the test cube and output response pairs in a way that minimizes the worst-case total number of channels. In this example, the worst-case pair requires 16 tester channels to decompress.

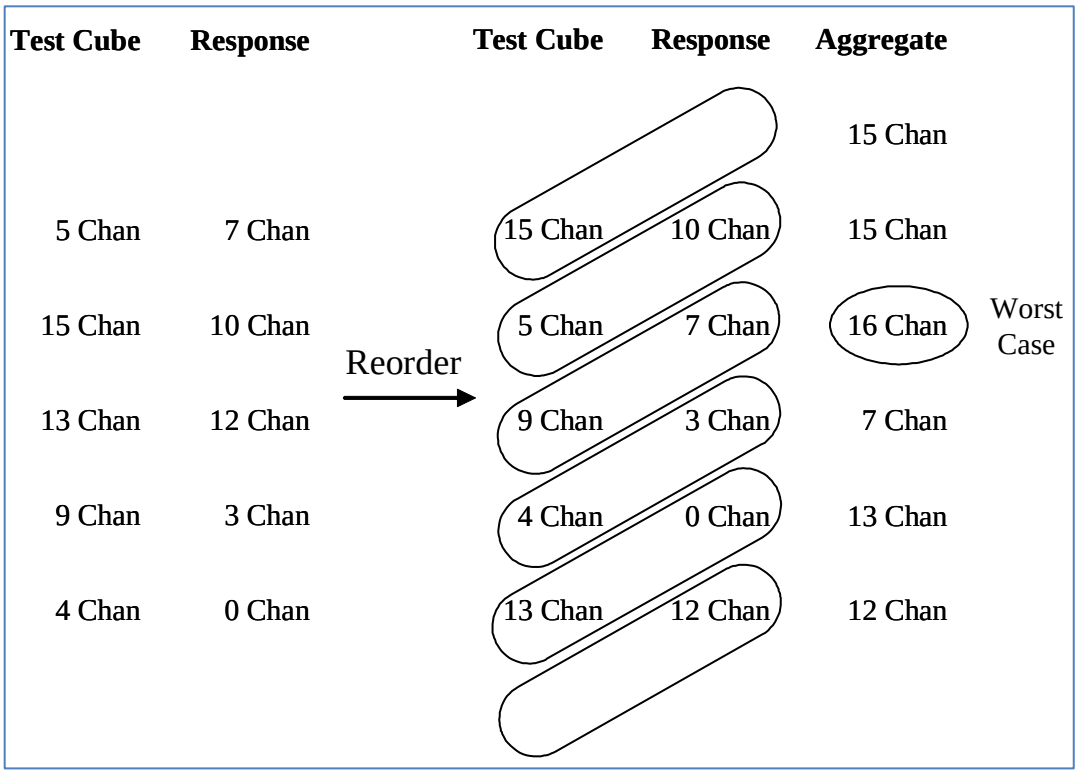


**Figure 12. Example of Reordering Test Cubes to Minimize Worst Case Total Number of Tester Channels**

Once the test cubes are ordered, the total number of channels needed to decompress each matched test cube and output response pair is computed by adding together the number of channels used for the test cube decompressor plus the minimum number of channels required for the mask decompressor. The maximum number of test channels required across all cases is the minimum number of tester channels that must be used when decompressing the test data for the CUT using this scheme.

## 3.5 Increasing Observability

As seen in the previous section, the total number of channels used depends on the worst-case across all matched test cubes and output response pairs. In many cases, the matched test cube and output response pair may require fewer channels. In those cases, more channels will be feeding the mask decompressor than necessary creating excess free variables. In this section, a technique for exploiting these excess free variables to improve observability is described based on the idea in [Volkerink 05] of ANDing together stages of the decompressor.

If two stages of the decompressor are ANDed together, the number of care bits needed to mask each X increases to 2 because the decompressor needs to generate 1's on both inputs of the AND gate in order to mask each X. However, for the non-X values, the probability of each being masked and losing observability is reduced to 0.25. If a three-input AND is used, then 3 care bits are needed to mask an X, and the probability of losing observability for a non-X is reduced to 0.125.

The idea proposed here is to add a control register, *m*, which controls how many stages of the decompressor are ANDed together to drive the mask logic. This is illustrated in Figure 13. The *m* register can be loaded at the same time as the *q* register in the first clock cycle for each test cube. In the example in Figure 13, the *m* register is a two-bit register which can take on 4 different values. This allows from 1 to 4 stages of the decompressor to be ANDed together to drive the mask logic. Depending on how many free variables are available in the mask decompressor, the *m* register can be set to AND as many stages as possible while still being able to solve the linear equations to encode the mask data.

If *y* channels are needed to encode the minimum mask data with no ANDing of decompressor stages, then the number of channels needed to encode the mask dating with ANDing of *k* stages can be estimated as *(k)( y)* since the number of care bits in the mask control data increases by a factor of *k*.

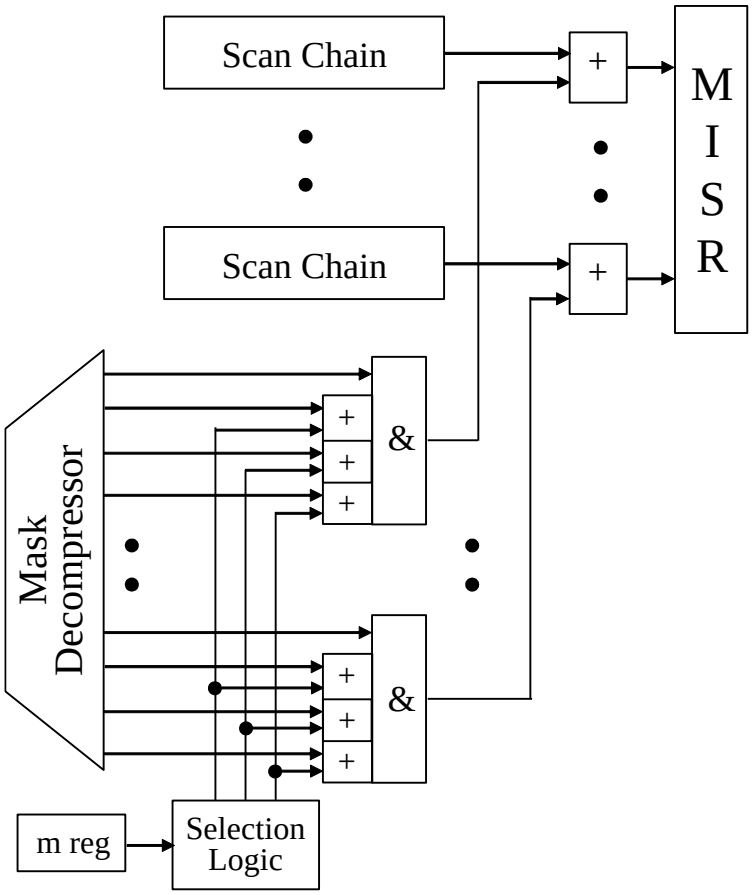


**Figure 13. Scheme for Selective ANDing of Multiple Outputs of Mask Decompressor to Improve Observability**

If the goal is to maximize the number of output responses that have $k>1$, then the procedure for ordering the test cubes is the following. Test cubes are selected in order from the one requiring the most channels to the one requiring the fewest channels. For each test cube, if all the remaining output responses that can be matched with it cannot have $k>1$ without exceeding the total number of test channels available, then the output response with the most channels that can be matched without exceeding the total number of test channels is selected. This increases the chance that subsequent test cubes can be matched with $k>1$.

Note that if the number of channels from the tester is increased beyond the minimum needed to encode all masks for *k=1*, then more output responses can be encoded with *k>1* thereby increasing observability. The tradeoff between increasing the number of channels versus the improvements in observability are explored in the experimental results presented in the next section.

## 3.6 Experimental Results

Experiments were performed on two industrial circuits to evaluate the proposed method of using dynamic channel allocation with test cube ordering. In Table 3, results are shown for the two circuits. The number of scan cells are shown for each circuit, followed by results for using the conventional approach of having a fixed number of tester channels driving the test cube decompressor and a fixed number of tester channels driving the mask decompressor. The amount of test data after compression is shown followed by the observability which in the conventional case is 50% assuming the decompressor drives the mask logic with an equal distribution of 0's and 1's. Next, results are shown for the proposed method. By dynamically allocating the channels, the total number of channels needed is reduced resulting in more compression. Results are shown for three different levels of observability. The first line for each circuit, shows the case where compression is maximized as much as possible and what the corresponding observability is. The next two lines show where the number of channels is increased above the minimum to increase the number of excess free variables thereby allowing greater use of higher values of *k* to improve observability. As can be seen from the results, a significant improvement in both compression and observability is achieved with the proposed method.

**Table 3. Experimental Results for Proposed Partial Masking in X-Chains for Different Designs**

| Circuit | Scan Chains | Fixed Channels | | Proposed Dynamic Channel Allocation | | | |
|---|---|---|---|---|---|---|---|
| | | Compressed Bits | Aggregate Observability | Compressed Bits | Percent Improve | Aggregate Observability | Percent Improve |
| Ckt-A | 36,075 | 19.3M | 50% | 11.2M | 42% | 64% | 28% |
| | | | | 13.4M | 31% | 70% | 40% |
| | | | | 15.7M | 19% | 77% | 54% |
| Ckt-B | 505,051 | 97.7M | 50% | 65.8M | 33% | 79% | 58% |
| | | | | 79.0M | 19% | 88% | 76% |
| | | | | 92.1M | 6% | 93% | 86% |

The tradeoff between compression and observability is explored further in the graphs in Figure 14 and Figure 15 for *Ckt-A* and *Ckt-B*, respectively. These graphs show the number of test cubes with different values of $k$ on the $y$-axis versus the increase in the number of tester channels over the minimum possible on the X-axis. As the number of tester channels are increased, the number of excess free variables increases allow more output responses to be shifted out with higher values of $k$ thereby increasing observability of the non-X values.

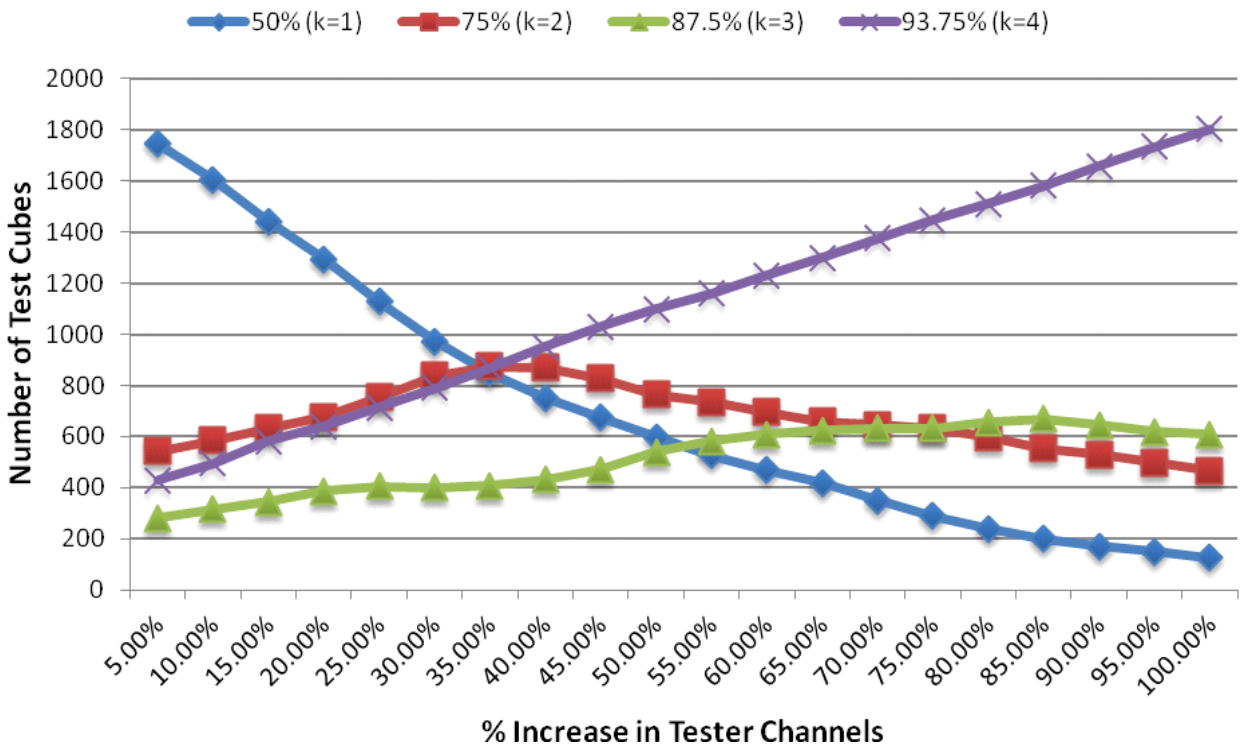


**Figure 14. Plot of Observability Improvement Versus Number of Tester Channels for Ckt-A**

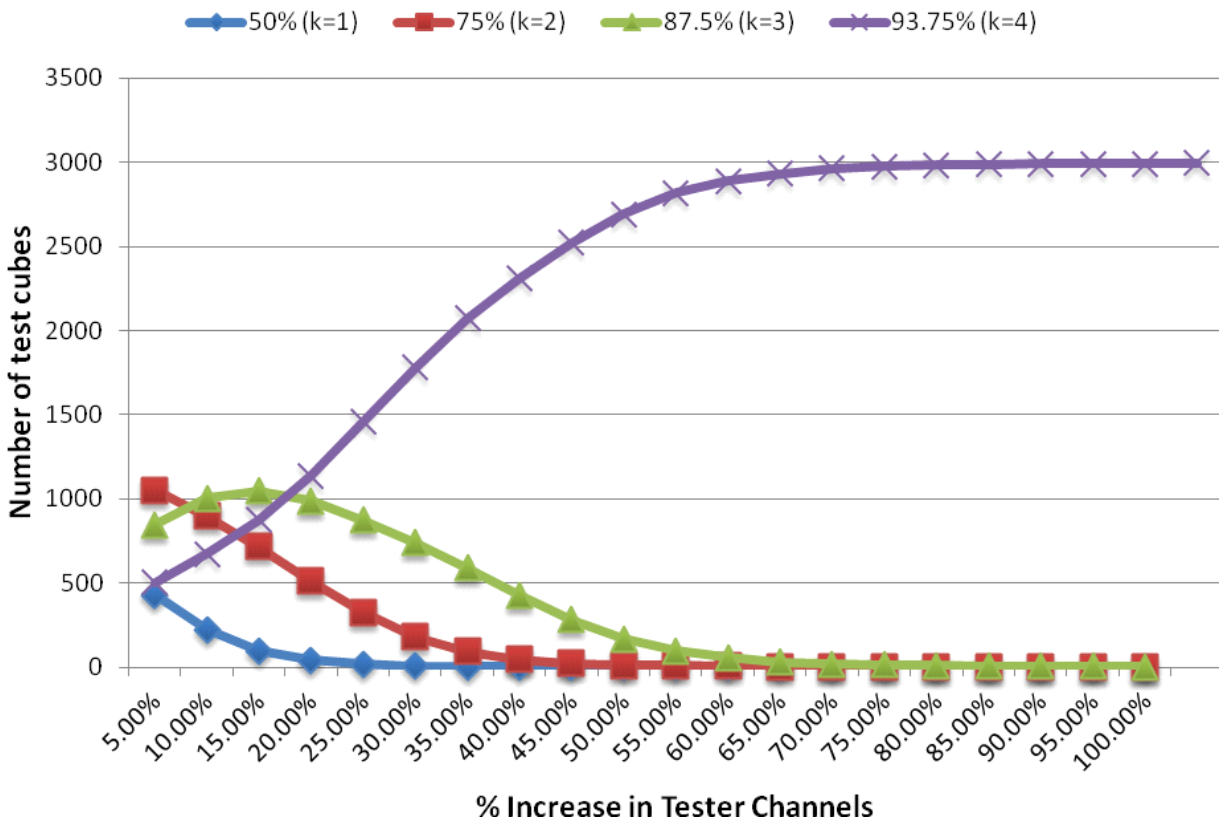


**Figure 15. Plot of Observability Improvement Versus Number of Tester Channels for Ckt-B**

As can be seen in Figure 14 *Ckt-A* starts off with most test cubes masked with *k=1* which then decreases almost linearly with more tester channels while the number of test cubes that can be masked with *k=4* increases almost linearly. In Figure 15, *Ckt-B* which has a lower X density, starts with most test cubes masked with *k>1* and gives a much higher observability for a small increase in tester channels. As the tester channels keeps increasing, a point is reached where all the test cubes are masked with *k=4*.

## 3.7 Summary

Dynamic channel allocation between the test cube decompressor and the output response decompressor can be implemented with a relatively small amount of logic (as illustrated in Figure 11), but can provide a significant boost in test compression. Moreover, higher observability of non-X values which is important for improving coverage of non-modeled faults can be achieved at no additional cost in terms of test data and relatively small additional hardware overhead (as illustrated in Figure 13) with selective ANDing of the mask decompressor outputs.

# Chapter 4: Improving X-Tolerant Combinational Output Compaction via Input Rotation

## 4.1 The Problem

The previous two chapters talked about different ways to handle X's in the output response. Similar to the X-cancelling MISR described in Chapter 1, an X-tolerant combinational compactor can compact an output stream that contains X's without the need for X-masking. Combinational linear compactors can be used to compact the output response for a large number of scan chains into a smaller number of outputs. While some compactor designs can guarantee observation of all scan chains in the presence of a small number of X's, this may not be sufficient for designs with higher X densities. This chapter describes an approach for using a combinational rotator between the scan chains and compactor to allow detection of faults even in the presence of high X densities.

## 4.2 Related Work

A number of techniques have been developed to handle X's in the output response. One approach is to modify the circuit- under-test (CUT) to eliminate the sources of X-values. This involves blocking sources of X within the circuit by inserting design-for-testability (DFT) hardware to prevent Xs from propagating into scan cells [Wang 06]. Another approach, which does not require modifying the CUT, is X-masking which masks out X's at the input to the compactor. Mask control data is used to specify which scan chain outputs should be masked during which clock cycles.

For low X-densities, techniques that use sequential linear compactors such as MISRs or convolutional compactors, can typically achieve higher amounts of compression than combinational compactors. However, because they XOR together large numbers of response bits, X values entering these compactors cause a greater degree of loss of non-X values which makes it harder for them to handle high X-densities. For high X-densities,

combinational compactors become more efficient in preserving observability thereby reducing test pattern inflation so as to achieve a higher overall amount of test compression. X-tolerant compactors have been developed based on linear combinational compactors [Mitra 04], [Patel 03], [Sharma 05] and [Wohl 07a, 07b, 08], finite memory compactors [Wang 03], [Rajski 05], [Rajski 06b] and [Gizdarski 10], and X-canceling MISRs [Yang 12], [Bawa 12] and [Chung 12].

The focus of this chapter is on combinational compactors which have the advantages of very simple design and low overhead. The original idea of designing a combinational linear circuit that can compact output responses with X's was first described by Mitra and Kim [Mitra 04]. The idea is that each scan chain fans out to multiple outputs where they are XORed together. The combinations of scan chains that are XORed together at each output are selected in such a way that if any single scan chain has an X value, it is still possible to observe all other scan chains. The scan chain with an X value will corrupt all the outputs that it fans out to which will then be masked on the tester, but all other outputs can still be observed as all other scan chains fan out to at least one of those outputs.

In [Wohl 07a], Wohl, et al., use the same principle, but limit the number of outputs that each scan chain fans out to only 3. The output compactor is designed using Steiner triple systems [Colbourn 99] such that each scan chain fans out to three outputs where no other scan chain fans out to more than one of those three outputs. Consequently, any two scan chains with X's can be tolerated while still allowing observation of all other scan chains.

While observation of all scan chains in combinational compactors can be guaranteed for a small number of X's using the methods in [Mitra 04] and [Wohl 07a], this may not be sufficient for designs with higher X densities. The scan cells that need to be observed to ensure detection of necessary faults for a test vector will be referred to as D values in this chapter. When the number of X's in a particular scan slice is sufficiently large, it may block observation of some D's. Handling more X's can be done through either masking [Wohl 07b] or filtering [Sharma 05]. The technique in [Wohl 07b] provides the

ability to selectively mask on a slice-by-slice basis a sufficient number of scan chains such that all unmasked scan chains can be directly observed through the compactor. The technique in [Sharma 05] involves adding an X-filtering circuit at the output of the compactor which can cancel out the X's. However, this comes at the cost of a large number of control inputs equal to a multiple of the number of X's to be tolerated per slice thereby making it unattractive.

The drawback of masking X's is that in order to keep the amount of data required for masking at a reasonable level, the number of different combinations of scan chains that can be masked has to be kept small. Consequently, the masking is coarse grain resulting in many non-X values also getting masked. This reduces the amount of observation and can result in more test patterns needing to be applied to achieve the same fault coverage (i.e., test pattern inflation).

The proposed idea involves tolerating high X-densities without masking. The key idea is to exploit the fact that for combinational compactors where the inputs fan out to a small number of outputs, a sizeable fraction of the inputs will remain observable even in the presence of many X's. For example, consider a compactor designed using Steiner triple systems using the procedure in [Wohl 07b] that has 610 inputs and 61 outputs. Even in the presence of 40 X's in a scan slice, over 35% of the inputs remain observable. Normally this is not sufficient as the probability of observing a particular D would be only 35%. However, the proposed approach adds a combinational rotator between the scan chains and the compactor which allows the connection of scan chains to compactor inputs to be shifted with the very last bit position being rotated to become the first bit position. By selectively rotating, a D can be matched to an observable input ensuring that it will be observed at the output of the compactor. A procedure for carefully ordering the inputs to the compactor to maximize the probability of having an observable input within a given maximum shift distance is described. Using this procedure, the number of control inputs required for the rotator can be minimized.

To illustrate the advantage of the proposed approach, consider the example mentioned earlier of a Steiner triple system compactor with 610 inputs and 61 outputs. Whereas a conventional masking technique that can directly observe any D will only observe 61 of the 610 inputs (i.e., 10% observability), the proposed approach using the same number of control inputs would also observe any D, but would provide much higher observability. As will be seen in the experimental results in Section 4.5 , for 10 X's, it would provide 95% observability, for 20X's, it would provide 76% observability, for 30X's it would provide 54% observability, and for 40X's it would provide 35% observability. Increased observability translates to less test pattern inflation and better compression as well as better coverage of non-modeled faults.

Another nice property of the proposed method is that the control inputs to the rotator are don't cares except for scan slices in which both D's and X's are simultaneously present. For an arbitrary scan slice with no D's or no X's, it doesn't matter how the inputs are connected to the compactor, the overall percentage of observable inputs will be approximately the same. This makes it very efficient to drive the control inputs to the rotator from a linear decompressor. This is a further advantage compared to masking approaches because driving the control inputs for masking circuitry with arbitrary values results in unnecessary masking and loss of observability, so typically an additional enable signal needs to be added to the design to disable the masking circuity when it is not needed if the other control signals are to be driven by a linear decompressor.

Note that the Response Shaper in [Chao 05] and X-Align in [Sinanoglu 09a, 09b] also involve adding a block between the scan chains and combinational compactor. However, these methods are fundamentally different from the proposed method. They involve using flip-flops to selectively delay subsets of scan chains so as to change the composition of X's and D's arriving at the inputs to the compactor. Whereas the proposed method is using purely combinational logic to rotate the inputs to the compactor and not changing the composition of X's and D's. Since it is only using combinational logic, the proposed method is a much simpler design with less overhead. The proposed method is

actually orthogonal to the methods in [Chao 05] and [Sinanoglu 09a, 09b] and could be used in conjunction with those methods.

## 4.3 Overview of proposed Scheme

A block diagram for the proposed scheme is shown in Figure 16. A combinational rotator is added between the scan chains and the inputs to a combinational compactor. The additional hardware required for the proposed scheme beyond what is already present for test compression is shown in red. The inputs to the combinational rotator can be driven by tester channels or can be driven by a linear decompressor. As mentioned earlier, these inputs are don't care except for scan slices when both D's and X's are simultaneously present. When both D's and X's are simultaneously present, the control signals to the combinational rotator are selected to propagate the D's.

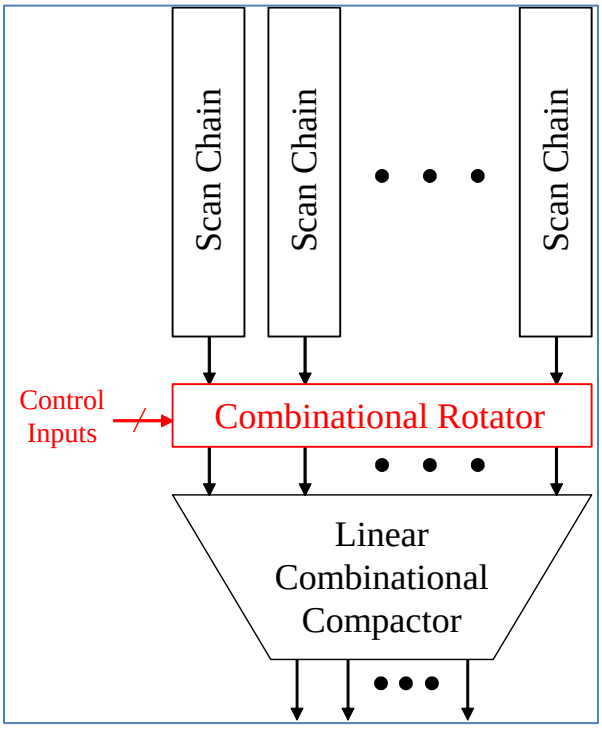


**Figure 16. Block Diagram of Proposed Scheme**

The number of control signals to the rotator determines the maximum shift distance that is possible. For *c* control inputs, the maximum shift distance would be $2^c$-1. The shift distance should be selected to achieve a high probability (e.g., greater than 99%) of being able to observe a D's under the expected X density. The number of control inputs will depend on the number of inputs and outputs to the compactor and the expected X density.

A small example illustrating how the scheme works is shown in Figure 17 and Figure 18 . There is one D input and two X inputs. In Figure 17 where the rotation is 0, the two compactor outputs that the D fans out to both receive X inputs as well which blocks observation at the output. In Figure 18, with the rotation set to 1, all the inputs are shifted one step to the left. In this case, the D fans out to one compactor output that does not receive any X inputs, so it is able to be observed. The addition of the combinational rotator provides the ability to avoid cases where D values get blocked.

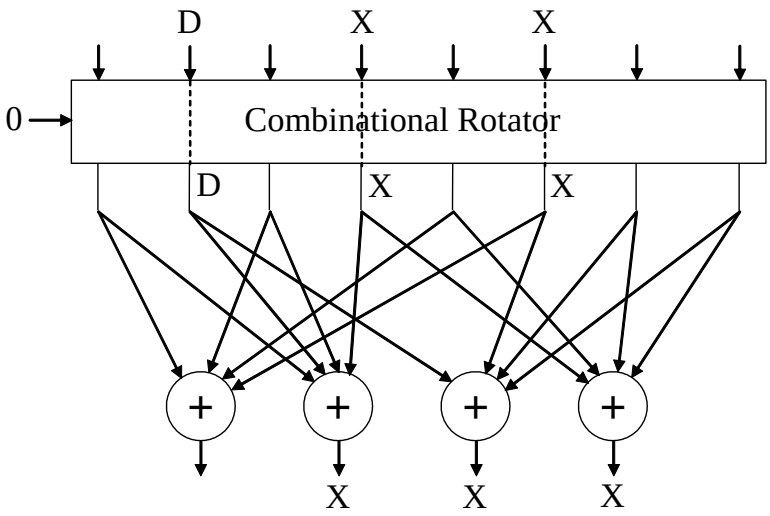


**Figure 17. Example of D Blocked from Observation**

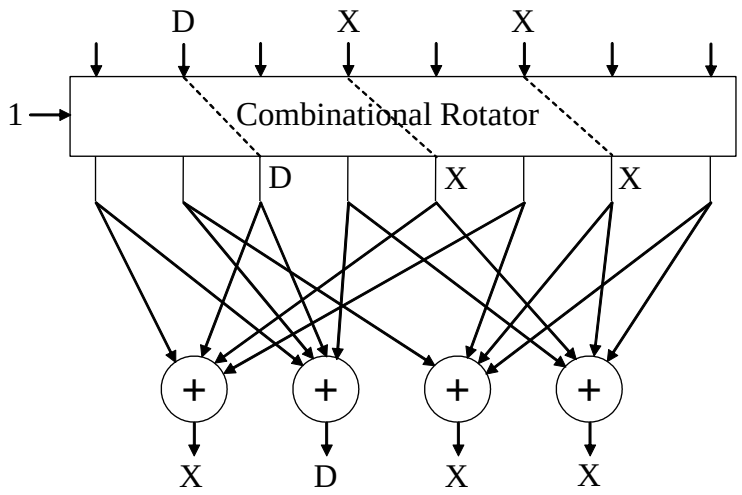


**Figure 18. D Becomes Observable with Rotation**

## 4.4 Procedure for Ordering Compactor Inputs

Since a combinational rotator only shifts inputs and doesn't permute them, the order of the inputs in the compactor design is important. Of course, for a completely symmetric compactor design, the order of the compactor inputs doesn't matter for the proposed scheme. However, in the general case, the compactor is not completely symmetric starting with the fact that the number of inputs times the number of fan outs per input may not necessarily be a multiple of the number of outputs. Even if that is the case, it would still not be symmetric unless the fan outs for every input are connected in the exact same pattern.

In the general case where the compactor is not symmetric, the probability of having an observable input within a given maximum shift distance depends on how the inputs to the compactor are ordered. Some input orders can be better than others. As a simple example, consider a maximum shift distance of 1. Assume X's are located at inputs *i* and *j* and a D is located at input *k*, and the X's block observation of the D at all outputs that the D fans out to (as in the example in Figure 17). If the inputs are shifted by a distance of 1,

then the X's will now be located at inputs $i$+1 and $j$+1, and the D will be located at input $k$+1. If an X at input $i$+1 and $j$+1 blocks a D at location $k$+1, then it will not be possible to observe the D within a maximum shift distance of 1. In order to maximize the probability that a D can be observed in the presence of two X's within a maximum shift distance of 1, the inputs should be ordered in a way that minimizes the number of $i, j, k$ sets where X's at $i$ and $j$ block a D at $k$, and X's at $i$+1 and $j$+1 block a D at $k$+1.

```
SHIFT_OBSERVE(p,offset) {
  For each output q reached from input position p {
    Initialize set REACH_q = ∅
    For each input position i that reaches output q {
      Add to set REACH_q all outputs reachable from i+offset
    }
  }
  If possible to cover all outputs reachable from p+offset
  by selecting one element from each set of REACH_q then {
    /* Some combination of X's that block p can block p+offset */
    Return(FALSE)
  }
  else {
    /* No combination of X's can block p+offset */
    Return(TRUE)
  }
}

HILL_CLIMB_INPUT_ORDERING() {
  Repeat {
    Initialize swap_performed = FALSE
    For each pair (p_1,p_2) of input positions for compactor {
      Initialize starting_value = 0
      If SHIFT_OBSERVE(p_1-1,offset) then starting_value++
      If SHIFT_OBSERVE(p_1,offset) then starting_value++
      If SHIFT_OBSERVE(p_2-1,offset) then starting_value++
      If SHIFT_OBSERVE(p_2,offset) then starting_value++
      Swap p_1 and p_2
      Initialize swapped_value = 0
      If ( SHIFT_OBSERVE(p_1-1,offset) ) then swapped_value++
      If ( SHIFT_OBSERVE(p_1,offset) ) then swapped_value++
      If ( SHIFT_OBSERVE(p_2-1,offset) ) then swapped_value++
      If ( SHIFT_OBSERVE(p_2,offset) ) then swapped_value++
      If ( swapped_value ≤ starting_value ) then {
        Swap p_1 and p_2 back to initial values
      }
      else if ( swapped_value > starting_value ) then {
        swap_performed = TRUE;
      }
    }
  } while ( swap_performed)
}
```

**Figure 19. Hill Climbing Procedure for Input Ordering**

Pseudo-code for a procedure that takes as an input a linear compactor design and uses a heuristic hill climbing process to reorder the inputs so as to improve the probability of observing D's through rotating inputs is shown in Figure 19. A basic subroutine used in the procedure is SHIFT_OBSERVE(*p*,*offset*) which determines whether a D at input position *p* that is blocked (where one X reaches each output that the D reaches) can be observed if it is shifted by some offset. The main procedure HILL_CLIMB_INPUT_ORDERING() is based on selecting a candidate pair of inputs ($p_1$,$p_2$) to swap. If swapping inputs $p_1$ and $p_2$ will increase the number of D positions that can be observed by a shift offset of 1, then the swap is performed. To determine this, four calls are made to SHIFT_OBSERVE, one each for $p_1$, $p_2$, $p_1$-1, and $p_2$-1. Those four positions are the only ones that can be affected for a shift offset of 1. The procedure could be expanded to also consider larger shift distances if desired. Since it is a hill climbing procedure, it can be terminated at any time, and the best solution found so far can be used. So, it is very easy to tradeoff runtime versus optimality of the result.

The heuristic procedure in Figure 19 is general can be used for any linear compactor design. Note that for specific classes of compactor designs, it may be possible to construct an optimal procedure for ordering the inputs.

## 4.5 Experimental Results

Experiments were performed for two different linear combinational compactor designs. Both were constructed using the procedure in [Wohl 07a] based on Steiner triple systems where each input to the compactor fans out to three outputs. One design compacts 425 scan chains into 51 outputs, and the other design compacts 610 scan chains into 61 outputs. Table 4 and Table 5 shows what percentage of D's could be observed for different percentage of X's using the conventional approach with no rotator, versus the proposed approach. Results are shown for different numbers of control inputs going to the rotator. The maximum shift distance for the rotator is $2^c$-1 where *c* is the number of control inputs. As can be seen from the tables, for both compactors, 5 control inputs were sufficient to

handle up to 10% X's while maintaining 99% observability of D's. As shown in Figure 20, for any given expected percentage of X's, the number of control inputs to the compactor can be selected accordingly in order to achieve a particular observability target. It can also be seen that as the number of inputs to the compactor goes up, the effectiveness of a particular number of control inputs slightly reduces. This is due to the fact that the ratio of the maximum shift distance for the rotator to the total number of inputs to the compactor is reducing.

The overhead for a combinational rotator is one MUX per control input for each scan chain. Since the number of control inputs can be expected to scale logarithmically as the number of scan chains increases, this overhead will scale well as the design size grows.

**Table 4. Percentage of D's Observed for Different Percentage of X's with Proposed Approach for 425-51 Compactor**

| Percent X's | 425-to-51 Compactor | | | | | |
|---|---|---|---|---|---|---|
| | No Rotator | Rotator (Control Inputs) | | | | |
| | | 1 | 2 | 3 | 4 | 5 |
| 0.5% | 100% | 100% | 100% | 100% | 100% | 100% |
| 1% | 99.6% | 100% | 100% | 100% | 100% | 100% |
| 2% | 95.1% | 100% | 100% | 100% | 100% | 100% |
| 3% | 86.2% | 98.2% | 100% | 100% | 100% | 100% |
| 4% | 78.9% | 94.3% | 99.7% | 100% | 100% | 100% |
| 5% | 64.8% | 87.8% | 98.5% | 100% | 100% | 100% |
| 6% | 54.2% | 79.1% | 95.7% | 99.9% | 100% | 100% |
| 7% | 44.4% | 69.2% | 90.5% | 99.1% | 100% | 100% |
| 8% | 33.9% | 56.5% | 81.1% | 96.4% | 99.9% | 100% |
| 9% | 27.0% | 46.8% | 71.7% | 92.0% | 99.4% | 100% |
| 10% | 20.1% | 36.2% | 59.3% | 83.5% | 97.3% | 99.9% |

**Table 5. Percentage of D's Observed for Different Percentage of X's with Proposed Approach for 610-61 Compactor**

| Percent X's | 610-to-61 Compactor | | | | | |
|---|---|---|---|---|---|---|
| | No Rotator | Rotator (Control Inputs) | | | | |
| | | 1 | 2 | 3 | 4 | 5 |
| 0.5% | 99.9% | 100% | 100% | 100% | 100% | 100% |
| 1% | 99.0% | 100% | 100% | 100% | 100% | 100% |
| 2% | 92.4% | 99.4% | 100% | 100% | 100% | 100% |
| 3% | 80.8% | 96.4% | 99.9% | 100% | 100% | 100% |
| 4% | 67.3% | 89.4% | 98.9% | 100% | 100% | 100% |
| 5% | 54.0% | 79.0% | 95.6% | 99.8% | 100% | 100% |
| 6% | 40.3% | 64.4% | 87.4% | 98.4% | 100% | 100% |
| 7% | 30.7% | 52.1% | 77.1% | 94.8% | 99.7% | 100% |
| 8% | 23.0% | 40.8% | 65.0% | 87.8% | 98.5% | 100% |
| 9% | 17.1% | 31.3% | 52.8% | 77.7% | 95.1% | 99.8% |
| 10% | 12.5% | 23.6% | 41.6% | 65.9% | 88.4% | 99.0% |

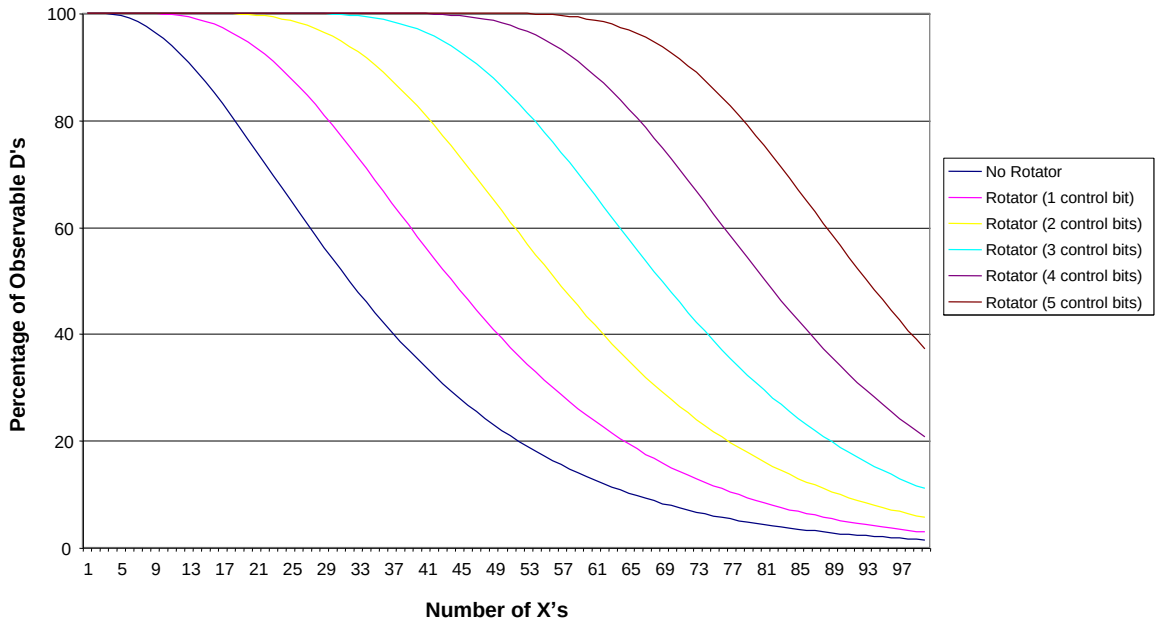


**Figure 20. Percentage Of D's Observed versus Number of X's per Slice for 610-to-61 Compactor**

## 4.6 Summary

The use of a combinational rotator is an attractive alternative to adding masking logic for handling designs with high X-density. While the number of control inputs and overhead is comparable to what would be required for masking logic, more scan cells get observed when no masking is performed. Moreover, the control inputs for a rotator will have more don't care conditions than masking logic which makes them more compressible if a decompressor is used to generate them. Note also that a rotator can be used in conjunction with other techniques such as Response Shaper [Chao 05], X-align [Sinanoglu 09], and even with masking techniques as well.

# Chapter 5: Algorithmic Design of Input Rotation Compactor for Improved Compaction

## 5.1 Background

The previous chapter introduced an approach for using a combinational rotator between the scan chains and compactor to allow detection of faults even in the presence of high X densities. In this chapter, a novel and completely new methodology for designing an output compactor is presented. It is based on using an input rotator that is able to maintain high observability even in the presence of high X-densities while still achieving high compaction ratios. The key idea is to place a combinational rotator in front of a carefully designed XOR network that maximizes separation of the input dependence in adjacent inputs within a particular shift distance of the input rotator. The amount of rotation can then be selected on a per cycle basis such that an output response bit that needs to be observed can be matched (with a very high probability) to an input to the XOR compactor that ensures that it will be observable after compaction.

## 5.2 Overview

The technique in the previous chapter places a combinational rotator between the scan chains and the compactor. The rotator allows the connection of scan chains to compactor inputs to be shifted with the very last bit position being rotated to become the first bit position. By selectively rotating, a D can be matched to an observable input ensuring that it will be observed at the output of the compactor. The approach in Chapter 4 was using a hill-climbing procedure in which inputs to a combinational compactor are swapped if the resulting compactor has a higher probability of observing D's through rotating inputs. The design in Chapter 4 uses a comparable number of control inputs and overhead to what would be required for masking logic, but more scan cells get observed thereby reducing test pattern inflation and therefore improving overall compression.

This chapter describes a completely new and novel methodology for designing an output compactor based on an input rotator that is able to maintain high observability even in the presence of high X-densities and high compression ratios. The key idea is to construct an XOR network in a way that maximizes separation of the input dependence in adjacent inputs within a particular shift distance of the input rotator. A systematic procedure is presented for constructing such a decompressor analytically given a targeted compression ratio and maximum shift distance. Using this procedure, the decompressor that is constructed maximizes the probability of observing a D in one of the inputs reachable within the maximum shift distance. Experimental results show significant improvement in observability for high X-densities in comparison to the compactor in Chapter 4 which translates to higher fault coverage, better diagnosis, and less overhead.

Note that the proposed method is compatible with methods that add flip-flops to selectively reshape output response such as Response Shaper in [Chao 05] and X-Align in [Sinanoglu 09a, 09b]. It can be used together with these methods, but for high X-densities it is effective by itself while using only combinational logic for a simpler design with less overhead.

## 5.3 Proposed Compactor Design

Since a combinational rotator only shifts inputs and doesn't permute them, the order of the inputs in the compactor design is important. The technique described here is to maximize the separation of bits that are compacted after rotation. The output of the combinational compactor can be represented using a dependency matrix where each row represents scan-chain output bits and columns represents the compactor output bits. The value '1' denotes that input is being compacted in the corresponding output bit and a '0' denotes the input is not in the fan-in cone of the output bit. Figure 21. shows a dependency matrix for a simple fanout=1 compactor, where every input is fanning out to just one output bit and each output bit is compacting three inputs e.g. $Z[0] = I[0] \oplus I[4] \oplus I[6]$.

In Figure 21 shifting of the inputs column down one row would simulate a shift distance of one and the result after shifting is shown on the right. As can be seen after shifting $Z[0]_{s1} = I[11] \oplus I[3] \oplus I[5]$

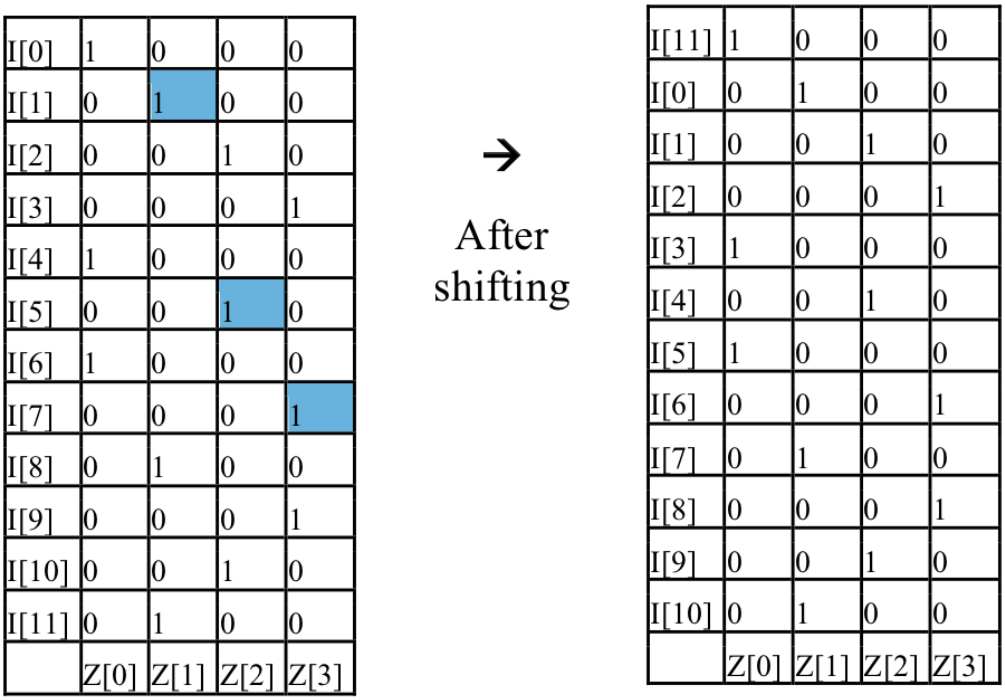


**Figure 21. Output Dependency Matrix**

In order to maximize the shift distance, the output dependency matrix has to be filled such that the bits that are going to a particular output before shifting go to different output bits after shifting. To achieve this all rows that have a '1' in the same column must have the very next row have '1' in different columns (column-staggering). In the example above, the rows after I[0], I[4] and I[6] have ones in different columns (blue fill is staggered). This is important to observe D's in the presence of X's, a simple example to demonstrate this is that if a D and X were being compacted in the same output bit before shifting they will be compacted by different output bits after shifting with a distance of 1.

It is also important to note that the dependency matrix maintains column-staggering regardless of the shifting distance applied.

The above example also shows that a compactor in which each input goes to only one output (i.e., fanout=1) with 3X compression needs at least 4 output bits so that each of the row after the three inputs compressed have column-staggering. This can be expanded to say that for a compressor with fanout=1 and n-output bits, the max compression is (n-1) X. Therefore, if a compression of 10X is required, it will need 11 output bits. This means compression can scale linearly with output bits and is only limited by the desired observability and X-densities.

## 5.4 Algorithm for Compactor Design

The algorithm described in the previous section is shown in Figure 22. The dependency matrix($i$,$j$) is set a row at a time. As the number of bits compacted for each output increases, the output($j$) tracks it and the value of j is incremented/rotated for the next row. The example shown in Figure 21 is generated by using the code below and works for any number of scan chains as long as $n$ output bits are used to achieve ($n$-1)X compression as described in the previous section

```
i = 0
j = 0
while {$i<$scan_chains} {
    matrix($i,$j) = 1

    if {output($j) == 0} {
        output($j) = 1
    } else {
        output($j) = output($j) + 1
    }

    i++
    j = $j + $output($j)
    j = $j % $outputs
}
```

**Figure 22. Pseudo Code**

## 5.5 Scaling to Higher Fanout

The compactor design technique shown in the previous sections can be easily modified for higher fanout. In general, fanout > 3 are not preferred due to excessive routing requirements. The Compactor shown in Figure 21 has multiple rows that are exactly the same and additional output *differential-bits* can be added to the compactor to differentiate these rows. The number of output bits needed will depend on the fanout and compression desired. Let's say the compactor shown in the previous section needs to be expanded for fanout=3. Since the compression is 3X, basically each row of the matrix shown in Figure 21 is exactly the same as two other rows and differential-bits need to be added to make them unique with the property that only two of the added bits can be a '1', this would ensure each row has exactly three '1's after two more '1's are added. Figure 23 shows the differential bits added to compactor shown in the previous section to make each row of the compactor matrix unique and different. It can be seen that rows 0, 4 & 6 were exactly the

same and each gets a different set of differential-bits. The total number of differential-bits needed below is 3 since ${}_3C_2$ is 3 which yields the three differential bit combinations of 011, 110 & 101. Each row in the matrix that has '1' in the same column then gets a different set of the of the differential bit. This can easily be expanded to higher fanout and compression, for example if a compression of 10X is desired with a fanout limit of 3, the number of differential bits needed will be 5 since ${}_5C_2$ will give 10 different differential-bit sets that can be added to each of the 10 rows that are compressed in the same output bit. The general formula for this is:

$$ {}_{(differential\text{-}bits)}C_{(fanout-1)} = Compression $$

As can be seen in Figure 23, the addition of differential bits significantly reduces the effective compression to 1.7X as 12 scan chains are now compacted into 7 output bits. This is not a problem for higher number of scan chains because, as noted previously, the differential-bits stays constant and only depends on fanout and desired compression. For example, if 1200 scan chains were compacted with 3X compressions, the number of output bits will be 400 bits to generate the fanout=1 matrix plus the 3 differential bits giving an effective compression of 2.98.

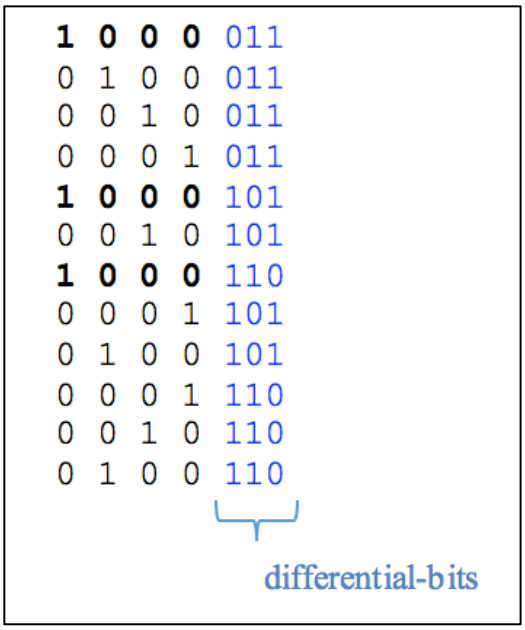


**Figure 23. Matrix with Differential-bits**

The technique described above generates an X-compact matrix that has the properties below and as described in [Mitra 04] is guaranteed to detect any one, two or odd number of errors in the same scan cycle in the absence of X's and will detect an error from any scan chain in the presence of one X without the need for any shifting.

- No row has all 0s
- Sub-matrix obtained by removing any row and the X-compact matrix columns having '1's in that row doesn't contain a row with all 0s
- Each row is distinct and contains an odd number of '1's

## 5.6 Experimental Results

Experiments were performed using a linear combinational compactor design that compacts 2420 scan chains into 121 outputs and the fanout of each scan chain is 3. The baseline was constructed using the procedure in [Wohl 07a] based on Steiner triple systems

where each input to the compactor fans out to three outputs with an input rotator as described in Chapter 4.

The proposed design is constructed with the fan-in limit of each output set to 22. Note that for 22X compression the output bits required for 2420 scan-chains is 110. In order to differentiate the rows that are the same in the fanout=1 matrix the minimal number of differential bits needed is 8 (since ${}_8C_2 = 28$ which is greater than 22). The total output bits would be 118 but in order to keep the scan chains and output bits constant, the initial fanout=1 matrix for the proposed approach is created with 113 outputs and with the 8 differential bits gets the total output bits to 121. Table 6 shows what percentage of D's could be observed for different percentages of X's using the conventional approach described in Chapter 4 versus the proposed approach. Results are shown for different numbers of control inputs going to the rotator. The maximum shift distance for the rotator is $2^c$-1 where c is the number of control inputs.

As can be seen from Table 6 which is plotted in Figure 24, the proposed approach gives a higher observability across the board and the proposed approach scaled a lot better as the percentage of X's increases which is clearly see in Figure 24 as the gap between the orange lines which is the proposed approach and the purple lines increase as the percentage of Xs increase. Also, once the X's percentage is above 3%, the new approach without rotator does better than the baseline with 2 rotator controls bits. Also, there is no overhead for the proposed approach when compared to baseline in Chapter 4.

**Table 6. Percentage of D's Observed for Different Percentage of X's**

| 2420-to-121 Compactor | | | | | | |
|---|---|---|---|---|---|---|
| %Xs | No Rotator | | 1 Rotator Control Bit | | 2 Rotator Control Bits | |
| | Baseline | Proposed | Baseline | Proposed | Baseline | Proposed |
| 0.50% | 86.5% | 90.6% | 98.9% | 99.2% | 100% | 100% |
| 1.00% | 64.4% | 82.0% | 88.6% | 96.8% | 98.8% | 99.9% |
| 1.50% | 42.8% | 74.5% | 68.3% | 93.2% | 90.3% | 99.6% |
| 2.00% | 26.8% | 67.5% | 46.9% | 89.3% | 72.1% | 98.9% |
| 3.00% | 9.8% | 55.7% | 18.7% | 80.0% | 34.0% | 96.1% |
| 4.00% | 3.5% | 46.1% | 6.8% | 69.7% | 13.2% | 90.7% |
| 5.00% | 1.4% | 39.0% | 2.7% | 61.1% | 5.4% | 84.2% |
| 6.00% | 0.6% | 33.8% | 1.2% | 52.4% | 2.4% | 76.4% |
| 7.00% | 0.3% | 29.1% | 0.5% | 45.9% | 1.1% | 68.7% |
| 8.00% | 0.1% | 25.2% | 0.3% | 40.1% | 0.5% | 60.7% |

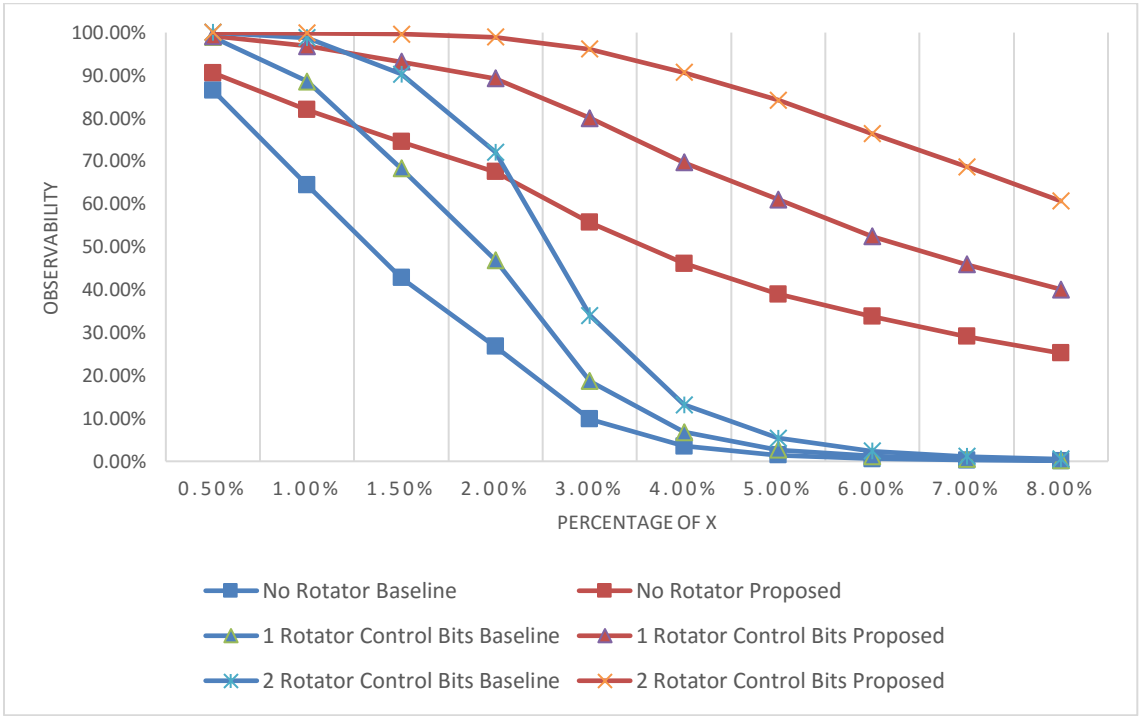


**Figure 24. Observability of D's for 2420-to-120 Compactor**

The experiment was repeated by using a higher compression for the proposed approach and the results are shown in Table 7 and plotted in Figure 25. In this case the fanout=1 matrix was created with 71 outputs and compression of 35X. This required 9 differential-bits that were used to differentiate each of the 35 rows that are the same. The effective compression for the new approach is 30X which is 50% higher than the baseline and as can be seen in Table 7 the proposed approach can easily be used to achieve higher compression in designs where the percentage X's are low. For example, if the design has 2% Xs and an observability of 90%+ is desired the new approach can be used to increase the compression instead of higher observability. Figure 25 also shows the same trend observed in Figure 24 which is the proposed approach does better with higher percentage of Xs which is clearly evident in the widening of the gap of the orange proposed approach to the purple baseline to the right of the chart.

The results shown in Figure 24 and Figure 25 clearly show that the proposed compactor uses the input rotator a lot more efficiently. This can be seen as the gap between the different orange lines for different rotator control bits is maintained as Xs increase in contrast to the purple lines that converge for X-densities above 4%.

**Table 7. Percentage of D's Observed for Different Percentage of X's with Higher Compression for Proposed Approach**

| 2420-to-80 Compactor | | | | | | |
|---|---|---|---|---|---|---|
| %Xs | No Rotator | | 1 Rotator Control Bit | | 2 Rotator Control Bits | |
| | Baseline | Proposed | Baseline | Proposed | Baseline | Proposed |
| 0.50% | 86.5% | 86.4% | 98.9% | 98.0% | 100% | 100% |
| 1.00% | 64.4% | 72.4% | 88.6% | 92.3% | 98.8% | 99.5% |
| 1.50% | 42.8% | 61.7% | 68.3% | 85.0% | 90.3% | 98.0% |
| 2.00% | 26.8% | 52.8% | 46.9% | 77.4% | 72.1% | 95.0% |
| 3.00% | 9.8% | 39.1% | 18.7% | 62.5% | 34.0% | 85.3% |
| 4.00% | 3.5% | 29.1% | 6.8% | 47.8% | 13.2% | 71.7% |
| 5.00% | 1.4% | 22.7% | 2.7% | 37.7% | 5.4% | 58.5% |
| 6.00% | 0.6% | 18.3% | 1.2% | 29.9% | 2.4% | 47.1% |
| 7.00% | 0.3% | 15.7% | 0.5% | 23.9% | 1.1% | 38.2% |
| 8.00% | 0.1% | 14.0% | 0.3% | 20.8% | 0.5% | 30.5% |

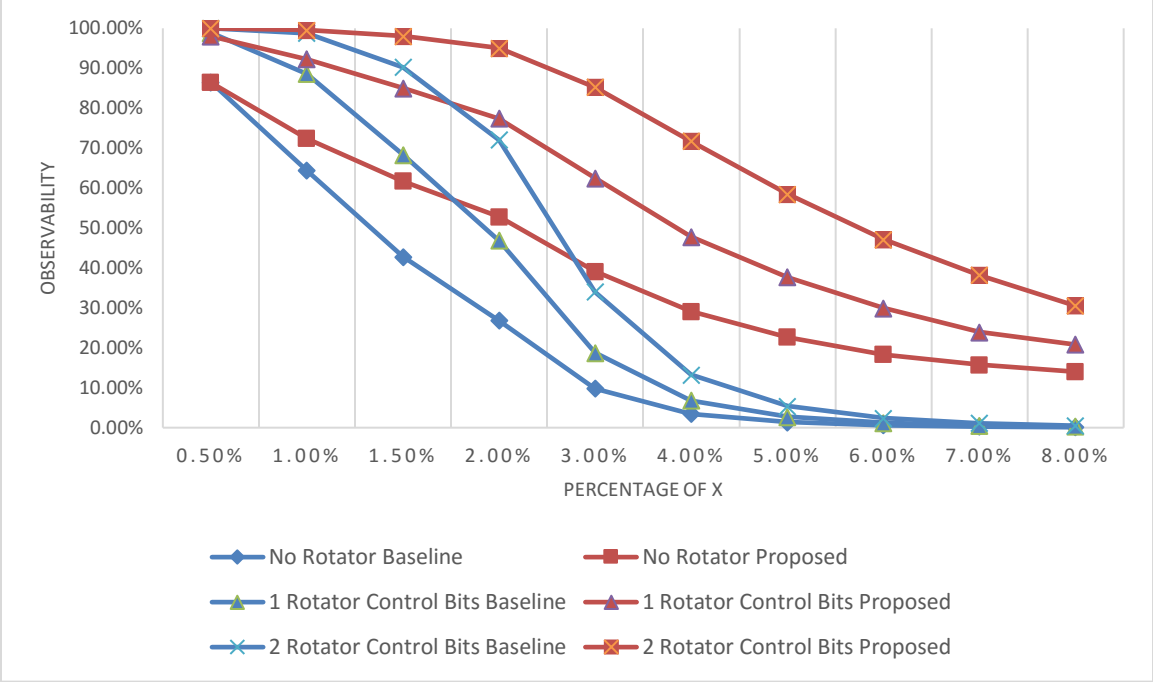


**Figure 25. Observability of D's for 2420-to-80 Compactor**

## 5.7 Summary

Using a combinational rotator is an attractive alternative to adding masking logic for handling designs with high X-density. By designing the improved technique described in this chapter, the combinational compactor with rotator is improved to scale to higher X percentages. Since the proposed technique gives better observability, it can also be used for designs with smaller X percentages by using a higher compression and reducing the output bits required thus reducing test time.

# Chapter 6: Conclusion and Future Work

This chapter summarizes the contributions of the dissertation in Techniques to Increase Compaction of Output Responses with Unknown (X) values and to enable higher observability for higher X-densities.

In Chapter 2, a technique is presented to combine partial X-masking and X-canceling to handle high X-densities. Because the proposed method can handle X-leaking, it is able to achieve high compression while still providing very precise masking where very little observation of non-X values is lost. This results in fewer test vectors and hence better test vector compression, output response compression, and test time. This work was published in [Bawa 12].

Chapter 3 presents the novel idea of using dynamic channel allocation between the test cube decompressor and output response decompressor and shows it can be implemented with a relatively small amount of logic. It is shown to provide a significant boost in test compression. Moreover, higher observability of non-X values which is important for improving coverage of non-modeled faults can be achieved at no additional cost in terms of test data and relatively small additional hardware overhead with selective ANDing of the mask decompressor outputs. This work was published in [Bawa 13]

In Chapter 4 a combinational rotator is presented as an attractive alternative to adding masking logic for handling designs with high X-density. The control inputs for a rotator will have more don't care conditions than masking logic which makes them more compressible if a decompressor is used to generate them. This work was published in [Bawa 15]

In Chapter 5 a novel combinational compactor is presented. The key idea is to construct an XOR network in a way that maximizes separation of the input dependence in adjacent inputs within a particular shift distance of the input rotator. This enables the combinational compactor with rotator to achieve higher compression and is improved to

scale to higher X-densities. This work was published in [Bawa 17] and won the Oscar W. Sepp Best Student Paper Award.

There are many areas for extending the current research work. X-masking techniques can be explored in combination with the combinational compactor with rotator described in Chapter 5. This can be investigated to scale to even higher X-densities or can be used to reduce test patterns. In addition, use of rotators can be explored in conjunction with other techniques such as Response Shaper [Chao 05] and X-align [Sinanoglu 09].

# References


[Barnhart 01] C. Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, B. Keller, and B. Koenemann, "OPMISR: the Foundation for Compressed ATPG Vectors," *Proc. of International Test Conference*, pp. 748-757, 2001.

[Bawa 12] A.A. Bawa, M. Tauseef Rab, and N.A. Touba, "Using Partial Masking in X-Chains to Increase Output Compaction for an X-Canceling MISR," *Proc. of IEEE Symp. on Defect and Fault Tolerance*, pp. 19-24, 2012.

[Bawa 13] A.A. Bawa, M. Tauseef Rab, and N.A. Touba, "Efficient compression of x-masking control data via dynamic channel allocation," *Proc. of IEEE Symp. on Defect and Fault Tolerance*, pp. 125-130, 2013.

[Bawa 15] A.A. Bawa and N.A. Touba, "Improving X-tolerant combinational output compaction via input rotation," *Proc. of IEEE Symp. on Defect and Fault Tolerance*, 2015.

[Bawa 17] A.A. Bawa and N.A. Touba, "Output Compaction for High X-Densities via Improved Input Rotation Compactor Design," *Proc. of IEEE Symp. AUTOTESTCON*, 2017.

[Chao 05] M. C.-T. Chao, S. Wang, S.T. Chakradhar, and K.-T. Cheng, "Response Shaper: A Novel Technique to Enhance Unknown Tolerance for Output Response Compaction," *Proc. of International Conference on Computer-Aided Design*, pp. 80-87, 2005.

[Chickermane 04] V. Chickermane, B. Foutz, and B. Keller, "Channel Masking Synthesis for Efficient On-Chip Test Compression," *Proc. of International Test Conference*, pp. 452-461, 2004.

[Chung 12] J. Chung and N.A. Touba, "Exploiting X-Correlation in Output Compression via Superset X-Canceling," *Proc. of VLSI Test Symposium*, pp. 182-187, 2012.

[Colbourn 99] C.J. Colbourn, and A. Rosa, "Triple Systems," Clarendon Press, Oxford, 1999.

[Cullen 97] C.G. Cullen, "Linear Algebra with Applications," Addison-Wesley, ISBN 0-673-99386-8, 1997.

[Czysz 10] D. Czysz, G. Mrugalski, N. Mukherjee, J. Rajski, and J. Tyszer, "On Compaction Utilizing Inter and Intra-Correlations of Unknown States," *IEEE Trans. on Computer-Aided Design*, Vol. 29, Issue 1, pp. 117-126, Jan. 2010.

[Datta 11] R. Datta, and N.A. Touba, "X-Stacking - A Method for Reducing Control Data for Output Compaction," *Proc. of IEEE Symposium on Defect and Fault Tolerance*, pp. 332-338, 2011.

[Gizdarski 10] E. Gizdarski, "Constructing Augmented Time Compactors," *Proc. of European Test Symposium*, pp. 151-156, 2010.

[Könemann 91] B. Koenemann, "LFSR-Coded Test Patterns for Scan Designs," *Proc. European Test Conf.*, pp. 237-242, 1991.

[Khoche 02]. A. Khoche, "Test resource partitioning for scan architectures using bandwidth matching," *Digest of Workshop on Test Resource Partitioning*, pp. 1.4.1-1.4.8, 2002.

[Krishna 01] C.V. Krishna, A. Jas, and N.A. Touba, "Test Vector Encoding Using Partial LFSR Reseeding," *Proc. Int. Test Conf.*, pp. 885-893, 2001.

[Mitra 04a] S. Mitra, and K.S. Kim, "X-Compact: An Efficient Response Compaction Scheme," *IEEE Trans. on Computer-Aided Design*, Vol. 23, No. 3, pp. 421-432, Mar. 2004.

[Mrugalski 04] G. Mrugalski, J. Rajski, and J. Tyszer, “Ring Generators – New Devices for Embedded Test Applications,” *IEEE Trans. on Computer-Aided Design*, Vol. 23, Issue 9, pp. 1306-1320, Sept. 2004.

[Mrugalski 09] G. Mrugalski, N. Mukherjee, J. Rajski, D. Czysz, and J. Tyszer, “Highly X-Tolerant Selective Compaction of Test Responses,” *Proc. of VLSI Test Symposium*, pp. 245-250, 2009.

[Naruse 03] M. Naruse, I. Pomeranz, S.M. Reddy, S. Kundu, "On-Chip Compression of Output Responses with Unknown Values Using LFSR Reseeding,” *Proc. of International Test Conference*, pp. 1060-1068, 2003.

[Patel 03] J.H. Patel, S.S. Lumetta, and S.M. Reddy, “Application of Saluja-Karpovsky Compactors to Test Responses with Many Unknowns,” *Proc. of VLSI Test Symposium*, pp. 107-112, 2003.

[Pomeranz 02] I. Pomeranz, S. Kundu, and S.M. Reddy, “On Output Response Compression in the Presence of Unknown Output Values,” *Proc. of Design Automation Conference*, pp. 255-258, 2002.

[Rajsi 02] J. Rajski, J. Tyszer, M. Kassab, N. Mukherjee, R. Thompson, K.-H. Tsai, A. Hertwig, N. Tamarapalli, G. Mrugalski, G. Eide, and J. Qian, "Embedded Deterministic Test for Low Cost Manufacturing Test," *Proc. of International Test Conference*, pp. 301-310, 2002.

[Rajski 05] J. Rajski, J. Tyszer, C. Wang, and S.M. Reddy, "Finite Memory Response Compactors for Embedded Test Applications," *IEEE Trans. on Computer-Aided Design*, Vol. 24, Iss. 4, pp. 622-634, Apr. 2005.

[Rajski 06a] J. Rajski, J. Tyszer, G. Mrugalski, W.-T. Cheng, N. Mukherjee, and M. Kassab, “X-Press Compactor for 1000x Reduction of Test Data,” *Proc. of International Test Conference*, Paper 18.1, 2006.

[Rajski 06b] W. Rajski, and J. Rajski, “Modular Compactor of Test Responses,” *Proc. of VLSI Test Symposium*, pp. 242-251, 2006.

[Rajski 08] J. Rajski, J. Tyszer, G. Mrugalski, W.-T. Cheng, N. Mukherjee, and M. Kassab, “X-Press: Two-Stage X-Tolerant Compactor with Programmable Selector,” *IEEE Trans. on Computer-Aided Design*, Vol. 27, Issue 1, pp. 147-159, Jan. 2008.

[Ramdas 12] A. Ramdas, and O. Sinanoglu, “Toggle-Masking Scheme for X-Filtering” *Proc. of European Test Symposium*, pp. 1-6, 2012.

[Sharma 05] M. Sharma, and W.-T. Cheng, “X-Filter: Filtering Unknowns from Compacted Test Responses,” *Proc. of International Test Conference*, Paper 42.1, 2005.

[Sinanoglu 09a] O. Sinanoglu and S. Almukhaizim, "X-Align: Improving the Scan Cell Observability of Response Compactors," *IEEE Trans. on VLSI*, Vol. 17, No. 10, pp. 1392-1404, Oct. 2009.

[Sinanoglu 09b] O. Sinanoglu and S. Almukhaizim, "X-alignment Techniques for Improving the Observability of Response Compactors,” *Proc. of International Test Conference*, Paper 17.1, 2009.

[Tang 06] Y. Tang, H.-J. Wunderlich, P. Engelke, I. Polian, B. Becker, J. Scholöffel, F. Hapke, and M. Wittke, “X-Masking During Logic BIST and Its Impact on Defect Coverage,” *IEEE Trans. on VLSI*, Vol. 14, No. 2, pp. 193-202 Feb. 2006.

[Touba 06] N.A. Touba, “Survey of Test Vector Compression Techniques,” *IEEE Design & Test Magazine*, Vol. 23, Issue 4, pp. 294-303, Jul. 2006.

[Touba 07] N.A. Touba, “X-Canceling MISR – An X-Tolerant Methodology for Compacting Output Responses with Unknowns Using a MISR,” *Proc. of International Test Conference*, paper 6.2, 2007.

[Volkerink 05] Volkerink, E.H., and S. Mitra, “Response Compaction with Any Number of Unknowns Using a New LFSR Architecture,” *Proc. of Design Automation Conference*, pp. 117-122, 2005.

[Wang 03] C. Wang, S.M. Reddy, I. Pomeranz, J. Rajski, and J. Tyszer, "On Compacting Test Response Data Containing Unknown Values," *Proc. of Int. Conf. on Computer-Aided Design (ICCAD)*, pp. 855-862, 2003.

[Wang 06] L.T. Wang, C.-W. Wu, X. Wen, “VLSI Test Principles and Architectures,” Morgan Kaufmann, 2006.

[Wang 08a] S. Wang, K.J. Balakrishnan, and W. Wei, "X-Block: An Efficient LFSR Reseeding-Based Method to Block Unknowns for Temporal Compactors," *IEEE Trans. on Computers*, Vol. 57, No. 7, pp. 978-989, July 2008.

[Wang 08b] S. Wang and W. Wei, "An Efficient Unknown Blocking Scheme for Low Control Data Volume and High Observability," *IEEE Trans. on Computer-Aided Design*, Vol. 27, No. 11, pp. 2039-2052, Nov. 2008.

[Whetsel 98] L. Whetsel, "Core test connectivity, communication, and control,” *Proc. of International Test Conference*, pp. 303-312, 1998.

[Wohl 01] P. Wohl, J.A. Waicukauski, and T.W Williams, “Design of Compactors for Signature-Analyzers in Built-In Self-Test,” *Proc. of International Test Conference*, pp. 54-63, 2001.

[Wohl 03] P. Wohl, J.A. Waicukauski, S. Patel, and M.B. Amin, “X-Tolerant Compression and Application of Scan-ATPG Patterns in a BIST Architecture,” *Proc. of International Test Conference*, pp. 727-736, 2003.

[Wohl 04] P. Wohl, J.A. Waicukauski, and S. Patel, “Scalable Selector Architecture for X-Tolerant Deterministic BIST,” *Proc. of Design Automation Conference*, pp. 934-939, 2004.

[Wohl 07a] P. Wohl, J.A. Waicukauski, R. Kapur, S. Ramnath, E. Gizdarski, T.W. Williams, and P. Jaini, "Minimizing the Impact of Scan Compression," *Proc. of VLSI Test Symposium*, pp. 67-74, 2007.

[Wohl 07b] P. Wohl, J.A. Waicukauski, and S. Ramnath, "Fully X-Tolerant Combinational Scan Compression," *Proc. of International Test Conference*, Paper 6.1, 2007.

[Wohl 08] P. Wohl, J.A. Waicukauski, and F. Neuveux, "Increasing Scan Compression by Using X-Chains," *Proc. of Design Automation Conference*, Paper 35.1, 2008.

[Wohl 10] P. Wohl, J.A. Waicukauski, and F. Neuveux, and E. Gizdarski, "Fully X-Tolerant, Very High Scan Compression," *Proc. of Design Automation Conference*, pp. 362-367, 2010.

[Yang 12] J.-S. Yang and N.A. Touba, "X-Canceling MISR Architecture for Output Response Compaction with Unknown Values," *IEEE Trans. On Computer-Aided Design*, Vol. 31, No. 9, pp. 1417-1427, Sept. 2012

# VITA

Asad Bawa completed his Bachelors in Electrical Engineering from The University of Texas at Austin in 2003. He then joined Magma Design Automation (presently Synopsys) as an Applications Engineer and continued at UT Austin working towards his Masters simultaneously. Upon completion of his Masters in Electrical and Computer Engineering (Circuit Design) in 2006, while working at Magma, Asad rejoined the University of Texas at Austin in 2008 to pursue a PhD. After having worked as an Application Engineer for 7 years, Asad joined Samsung Research Center in Austin, TX as a Physical Design Engineer in 2010. After a year of employment, he joined Apple Inc. in 2011, also as a Physical Design Engineer, and is currently leading/managing a physical design team working on GPUs.

Address: bawa@utexas.edu

This manuscript was typed by the author.